Títol: Anàlisi, disseny i implementació d'un sistema de control d'accés a la xarxa cablejada de la FIB

Autor: Oriol Bellet Chacón

Data: Octubre del 2011

Director: Daniel Sánchez Dorado

Ponent: Jaime María Delgado Merce

Departament del ponent: Arquitectura de Computadors (AC)

Titulació: Màster en Tecnologies de la Informació (MTI)

Centre: Facultat d'Informàtica de Barcelona (FIB)

Universitat: Universitat Politècnica de Catalunya (UPC) - BarcelonaTech.

Índex

1. Introducció

Actualment, les organitzacions disposen d'un gran nombre de dispositius (ordinadors de sobretaula, portàtils, impressores, etc.) a través dels quals els seus membres accedeixen a la xarxa. Aquestes organitzacions han de vetllar perquè aquest accés es realitzi de forma segura, sense comprometre el seus sistemes. Aquest augment en la seguretat, però, no hauria de suposar un increment de la dificultat de l'usuari per accedir-hi.

Aquest projecte sorgeix de la necessitat de crear un sistema que permeti als estudiants i altres membres de la Facultat Informàtica de Barcelona accedir a la xarxa cablejada de la universitat d'una forma còmoda i flexible, garantint, de principi a fi, la seguretat en les comunicacions.

Antigament, la seguretat en les xarxes cablejades era un problema menystingut. Però el gran augment del nombre d'usuaris que disposen d'un ordinador portàtil capaç de ser connectat a la xarxa ha fet replantejar aquest problema i entendre la necessitat de dotar la xarxa d'un cert nivell de seguretat.

Aquest projecte pretén estudiar la viabilitat de la implantació d'un sistema de control d'accés a la xarxa cablejada de la Facultat Informàtica de Barcelona.

1.1 El Laboratori de Càlcul de la Facultat Informàtica de Barcelona (LCFIB)

El Laboratori de Càlcul és l'agent encarregat de prestar serveis a la docència i administració de la FIB, facilitant els recursos humans i materials necessaris per proporcionar un servei de qualitat als seus usuaris. Actualment l'LCFIB està estructurat en les següents àrees de treball:

- Àrea de serveis i Xarxes de Comunicacions.
- Àrea de Sistemes d'Informació i Gestió.
- Àrea de Desenvolupament de Projectes Tecnològics.

L'Àrea de Serveis i Xarxes de Comunicacions és qui gestiona tot el sistema de comunicacions de la facultat. Això inclou tant les infraestructures (xarxes, electricitats, SAIs, CPDs, etc.) com els serveis que dóna la facultat (web, correu electrònic, estacions de treball, etc.). La xarxa física, motiu d'aquest projecte, està formada per 1138 punts d'accés a través dels quals els usuaris accedeixen a la xarxa, i per 22 commutadors i encaminadors encarregats de gestionar

les dades que hi circulen per ella. Tots aquests dispositius hauran de treballar conjuntament per formar el sistema de control d'accés.

1.2 L'actual control d'accés a la xarxa cablejada de la FIB

Actualment, el control d'accés a la xarxa cablejada de la facultat és realitza a nivell de dispositiu. Cada port físic de cada commutador té assignat un únic dispositiu al qual pot oferir accés a la xarxa. Per tant, cada dispositiu que vulgui accedir a la xarxa només ho podrà fer a través del port físic del commutador que tingui associat.

Per comprovar si a un port s'hi ha connectat el dispositiu correcte, els commutadors disposen d'una relació entre cada port físic i l'identificador únic de la targeta de xarxa del dispositiu que s'hi pot connectar. El commutador consulta aquesta informació cada cop que detecta que un dispositiu s'ha connectat a un dels seus ports. Si la comprovació és vàlida, el commutador permetrà l'accés del dispositiu a la xarxa. En cas contrari, l'accés serà denegat i el port quedarà bloquejat, evitant qualsevol tipus d'accés.

Aquesta implementació ofereix una sèrie de limitacions:

- Un dispositiu no pot ser connectat a cap port físic que no sigui el que se l'hi ha assignat. La conseqüència d'això és que quan un ordinador s'espatlla i ha de ser substituït per un altre, o la seva targeta de xarxa deixa de funcionar i ha de ser canviada, obliga a l'LCFIB a actualitzar les dades que consulta el commutador per validar el dispositius. Aquesta tasca es realitza manualment.
- No poden accedir a la xarxa dispositius dels quals el commutador no en conegui la seva existència. Per exemple, un estudiant no pot accedir a la xarxa cablejada a través del seu ordinador portàtil personal.
- El control d'accés es basa únicament en el dispositiu. Això provoca que els privilegis de xarxa no es puguin assignar en funció de l'usuari, ja que no es pot saber quin usuari està connectat a la xarxa.
- Al no conèixer qui està connectat, no es pot obtenir un registre d'activitat ni d'accés per usuari, ni saber quin ús de la xarxa n'està fent.
- Quan a un port s'hi connecta un dispositiu no autoritzat, el port queda bloquejat. La tasca de desbloquejar el port per a que torni a estar actiu l'ha de realitzar un membre del Laboratori de Càlcul manualment.

1.3 Objectius del projecte

L'objectiu del projecte és estudiar la viabilitat de la implantació d'un sistema de control d'accés a la xarxa flexible i potent que solucioni totes les limitacions que ofereix el sistema actual. Els requisits d'aquest nou control d'accés són:

- **Autenticació d'usuaris** Aquest és el principal motiu que impulsa a implantar un sistema de control d'accés. El commutador no només haurà de comprovar l'identificador de la targeta de xarxa del dispositiu per decidir si se li dóna accés a la xarxa o no, sinó que l'autenticació també s'haurà de basar en la identitat de l'usuari, que haurà de facilitar unes credencials què el sistema haurà de validar abans de permetre-li l'accés a la xarxa.
- **Definir els recursos que s'oferiran a cada usuari** Cada usuari i dispositiu del sistema haurà de pertànyer a un perfil. El recursos de xarxa que se li oferiran a cada usuari hauran de dependre del seu perfil o del perfil del dispositiu des del qual està connectat.
- **Mantenir un registre d'activitat de cada usuari** El sistema haurà de ser capaç de generar i emmagatzemar dades estadístiques sobre l'ús que cada usuari fa dels recursos de xarxa que se li han facilitat.
- **Monitoritzar el comportament del sistema** El sistema haurà d'estar controlat per un entorn de centralització i anàlisi de *logs*. D'aquesta forma es podran detectar caigudes dels servidors i comportaments anòmals.

El projecte es divideix en quatre fases ben diferenciades que s'aniran desglossant al llarg del document:

- **Recerca** En aquesta fase s'investigaran les diferents tecnologies existents que podrien satisfer els requisits i objectius del projecte.
- **Disseny** En aquesta fase s'analitzaran els resultats obtinguts en la recerca i es dissenyarà el nou sistema de control d'accés.
- **Implementació** En aquesta fase s'implementarà un pilot del sistema a partir del disseny resultant de la fase anterior.
- **Proves** En aquesta fase es descriuran tots els testos realitzats per verificar el correcte funcionament del pilot.

1.4 Planificació inicial

A continuació es mostra la planificació temporal del projecte. La seva data d'inici prevista és l'1 de Març del 2011 i la de finalització el 30 de setembre del mateix any. La durada total és de 128 dies, que equivalen a 1024 hores.

En el càlcul de dies no s'han comptabilitzat ni els caps de setmana, ni els dies festius (22 i 25 d'Abril, 24 de Juny), ni el mes d'Agost (període de vacances). El temps invertit en cada dia treballat és de 8 hores.

La planificació s'ha realitzat en funció de les quatre fases en les que es divideix el projecte. Com que a l'inici del projecte es desconeixien les tecnologies a aplicar i les possibilitats que podien oferir per assolir els objectius del projecte, ha estat impossible dividir el projecte en etapes més concretes. En la figura 1.1 és mostra el diagrama de Gantt resultant de la planificació inicial.

La primera de les etapes a realitzar és la de recerca. Un cop finalitzada es podrà començar la de disseny, que serà la predecessora de les etapes d'implementació i proves (que es realitzaran simultàniament). L'última de les etapes és la de Documentació, que es realitzarà durant tot el projecte (tot i que se'n dedicaran més esforços al final).

La taula 1.2 mostra més detalladament la planificació inicial. Cadascuna de les etapes té una data d'inici i final, el nombre de dies treballats que es dedicaran a l'etapa, una dedicació (percentatge de temps que es dedica a la tasca quan aquesta està en fase de realització), una duració efectiva (temps dedicat exclusivament a l'etapa) en dies i hores, i el percentatge de temps dedicat a cadascuna d'elles.

Com s'observa, l'etapa de Proves i Documentació s'ha dividit en dues parts. Això es degut a que la dedicació a cadascuna d'aquestes etapes varia al llarg del projecte.

1.5 Consideracions

Al llarg del document, aniran apareixent termes en anglès. S'ha decidit no traduir alguns d'ells per tal de mantenir les nomenclatures originals, que en molts casos són estàndards. En altres casos, l'absència d'un mot equivalent en llengua catalana ha fet impossible la seva traducció. Totes les paraules no traduïdes es podran identificar fàcilment perquè estan escrites en *cursiva*.

En molts casos també s'utilitzaran acrònims. Al final del document s'adjunta un glossari on es mostra el significat de cada cadascun d'ells.

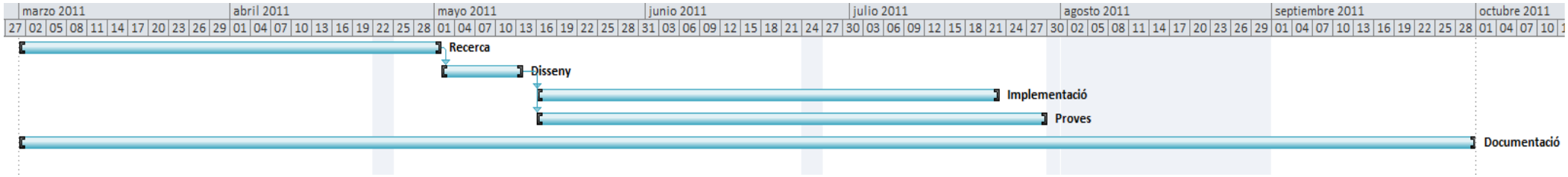


Figura 1.1 Planificació inicial del projecte

Etapa	Data Inici	Data Fi	Duració (dies)	Dedicació	Duració efectiva (dies)	Duració efectiva (hores)	% temps dedicat
Recerca	01/03/2011	01/05/2011	43	90%	38.7	309.6	30.324%
Disseny	02/05/2011	15/05/2011	10	90%	9	72	7.031%
Implementació	16/05/2011	24/05/2011	49	65%	31.85	254.8	24.883%
Proves (1)	16/05/2011	24/05/2011	49	25%	12.25	98	9.570%
Proves (2)	25/07/2011	31/07/2011	5	90%	4.5	36	3.516%
Documentació (1)	01/03/2011	31/07/2011	107	10%	10.7	85.6	8.360%
Documentació (2)	01/09/2011	30/09/2011	21	100%	21	168	16.406%

Taula 1.2 Planificació sobre la dedicació a cada etapa del projecte

2. Estat de l'art

En aquest capítol es descriu la fase de recerca del projecte. En ell s'estudien les possibles formes en les que es pot implantar un sistema de control d'accés a la xarxa.

El control d'accés fa referència a aquell conjunt de normes o accions que permeten incrementar el grau de seguretat en la xarxa. Algunes de les funcionalitats més comuns que es poden obtenir gràcies a un control d'accés són:

- Denegar l'accés als usuaris no autoritzats.
- Limitar els recursos als quals es poden accedir.
- Comprovar qui es va connectar, quan ho va fer i des de quin punt d'accés.
- Saber qui està connectat a la xarxa en temps real.

En un control d'accés hi intervenen bàsicament tres agents:

- **Usuari** És la persona que vol obtenir accés a la xarxa.
- **Servidor** S'encarrega de validar la identitat de l'usuari i decidir quins tipus de recursos de xarxa se li oferiran.
- **Commutador** És qui permet a l'usuari accedir a la xarxa en funció de les decisions presses pel servidor.

Com s'anirà observant al llarg del capítol, cada tecnologia anomenarà aquests tres agents de forma diferent. Cal notar que tot i aquesta falta de conveni, el rol que juguen cadascun d'ells és sempre el mateix.

Per altra banda, en el control d'accés hi ha dos tipus de comunicacions en funció dels agents que hi intervenen:

- **Control d'accés basat en ports** Comunicació entre l'usuari i el commutador.
- **AAA** Comunicació entre el commutador i el servidor.

A mesura que es vagi avançant en el capítol s'anirà profunditzant sobre cadascun dels conceptes aquí introduïts. De moment, es mostra una figura que simbolitza la distribució dels diferents agents en la xarxa i els tipus de comunicació utilitzen entre ells:

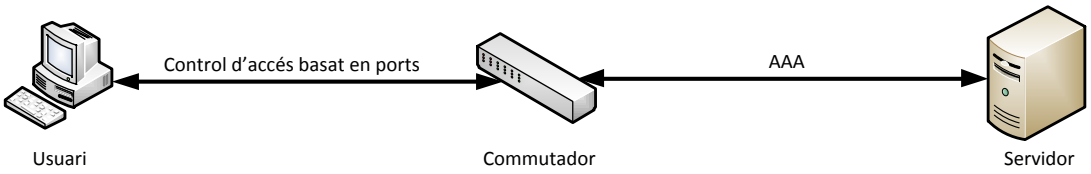


Figura 2.1 – Esquema de xarxa d'un control d'accés

2.1 Control d'accés basat en ports

El control d'accés (o autenticació) basat en ports utilitza les característiques físiques d'una infraestructura de xarxa d'àrea local commutada per autenticar els dispositius connectats a un port i impedir-ne l'accés quan es produeixi un error en el procés d'autenticació [1].

En el control d'accés basat en ports estan involucrats els tres agents descrits anteriorment:

- ***Supplicant* (usuari)** És el dispositiu que vol obtenir accés a la xarxa. És comunica físicament amb l'*Authenticator*.
- ***Authentication Server (*servidor)** És el servidor AAA encarregat d'autenticar l'usuari i decidir quins tipus de recursos de xarxa se li oferiran. Es comunica físicament amb l'*Authenticator*.
- ***Authenticator* (commutador)** Es troba situat entre el *Supplicant* i l'*Authentication Server*. És qui posseeix el port el qual vol ser utilitzat pel *Supplicant*. La seva funció és fer d'interlocutor entre el *Supplicant* i l'*Authentication Server* per permetre la comunicació lògica en ells. L'*Authenticator* també s'encarrega de gestionar els recursos de xarxa oferts al *Supplicant*.

La següent figura mostra la distribució dels tres agents a la xarxa:

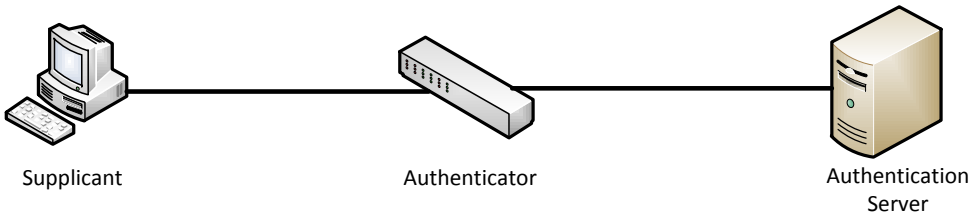


Figura 2.2 – Esquema de xarxa d'un sistema de control d'accés basat en ports

Tots els agents es comuniquen entre ells, ja sigui d'un forma física o lògica:

- ***Supplicant – Authenticator*** Es comuniquen físicament.

- ***Authenticator – Authentication Server*** Es comuniquem físicament.
- ***Supplicant – Authentication Server*** Es comuniquen lògicament.

En aquest apartat dedicat al control d'accés basat en ports s'analitzen els principals protocols utilitzats per dur a terme la comunicació física entre *Supplicant* i l'*Authenticator*: 802.1x, VMPS i MAB.

2.1.1 IEEE 802.1x

802.1x és un protocol creat per l'IEEE per al control d'accés a la xarxa basat en ports [1]. Aplica *l'Extensible Authentication Protocol* (*EAP)* sobre la xarxa d'àrea local a través de *l'Extensible Authentication Protocol Over LAN* (*EAPOL*). El protocol 802.1x és utilitzat per transportar les credencials des del *Supplicant* fins a l'*Authenticator*.

Així, l'autenticació 802.1x és un conjunt de trames EAPOL, que a la vegada són un tipus de trames EAP. La informació d'autenticació es troba en els anomenats EAP-Methods, que no són més que els camps de dades de les trames EAP. La següent figura mostra els diferents encapsulaments:

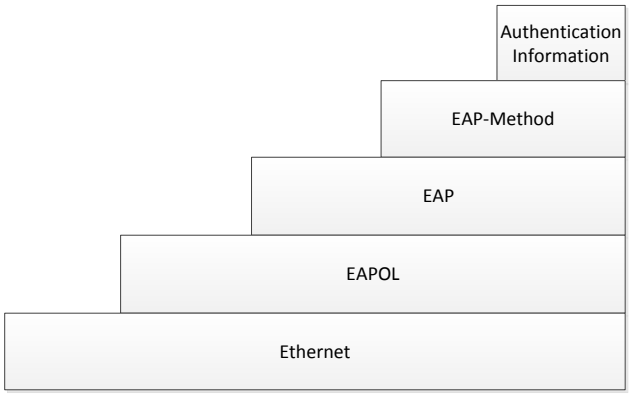


Figura 2.3 – Pila d'encapsulaments 802.1x

El protocol 802.1x impedeix que un usuari accedeixi als recursos de xarxa sense haver-se autenticat prèviament. Les credencials de l'usuari són encapsulades en una trama EAPOL i enviades a l'*Authenticator*, que desencapsula les dades i les reenvia (encapsulades de nou) a l'*Authentication Server*, que serà qui finalment realitzarà l'autenticació de l'usuari. Mentre l'usuari no estigui autenticat a la xarxa, l'*Authenticator* només acceptarà l'enviament de trames EAPOL per part del *Supplicant*.

2.1.1.1 EAP Over LAN (EAPOL)

L'IEEE van definir l'encapsulament EAPOL en les especificacions del protocol 802.1x degut a que EAP no oferia funcionalitat LAN.

EAP va ser creat per treballar sobre enllaços sèrie (com per exemple PPP). En aquest tipus de situacions, un enllaç es pot trobar o bé actiu o bé inactiu. Quan el *Supplicant* connecta el dispositiu l'enllaç s'activa, i quan el desconnecta és desactiva.

Quan es treballa en xarxes d'àrea local, aquesta situació no és així. Un *Supplicant* podria connectar un dispositiu intermedi, com per exemple un telèfon IP, i no provocar una activació de l'enllaç (tot i que el *Supplicant* seguiria necessitant autenticació).

EAPOL permet, a un dispositiu que necessita autenticació, enviar a l'*Authenticator* una petició d'iniciació del procés d'autenticació.

2.1.1.1.1 Trames EAPOL

A continuació es mostra el format de les trames EAPOL i es descriuen els seus camps [1]:

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
Version	Type	Length
EAP Data …		

Taula 2.4 – Format d'una trama EAPoL

- **Version** Indica la versió del protocol. Actualment la 2.
- **Type** Representa el tipus de trama EAPOL:

  0 - EAP Data

  Conté una trama EAP. La majoria de trames EAPOL seran d'aquest tipus.

  1 - EAPOL-Start

  Aquest tipus de trama es enviada pel *Supplicant* amb l'objectiu d'indicar-li a l'*Authenticator* que ha d'iniciar el procés d'autenticació.

  2 - EAPOL-Logoff

  Aquest tipus de trama es enviada pel *Supplicant* amb l'objectiu d'indicar-li a l'*Authenticator* que ha acabar d'utilitzar la xarxa, i que ja port tornar a col·locar el port en el seu estat inicial.

  3 - EAPOL-Key

Utilitzada per intercanviar informació sobre claus de xifrat entre el *Supplicant* i l'*Authenticator*. Actualment només s'utilitza en comunicacions sense fils.

4 - EAPOL-Encapsulated-ASF-Alert

Permet que el *Supplicant* enviï alertes del tipus *Alerting Standards Forum* en protocols com SNMP, que podran ser enviades a través del port encara que el *Supplicant* no hagi estat autenticat.

- **Length** Longitud del camp *EAP Data*. Val 0 quan el cos del missatge és buit.
- **EAP Data** Conté la trama EAP. S'assumirà que tota dada més gran que el valor del camp *Length* forma part del *padding* i serà ignorada pel receptor.

2.1.1.2 Extensible Authentication Protocol (EAP)

EAP és un *framework* que suporta múltiples mètodes d'autenticació. Funciona directament sobre la capa d'enllaç del model OSI com el Protocol *Point-to-Point* (PPP) o IEEE 802, sense requerir IP.

La principal funció de EAP és gestionar i transportar la informació (EAP-Methods) entre el *Supplicant* i l'*Authentication Server*. Això inclou la negociació de l'EAP-Method ha utilitzar en la comunicació, l'intercanvi de credencials i la confirmació sobre si l'autenticació s'ha realitzat correctament o no.

2.1.1.2.1 Trames EAP

A continuació es mostra el format de les trames EAP i es descriuen els seus camps [1]:

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
Code	Identifier	Length
EAP Data …		

Figura 2.5 – Format d'una trama EAP

- **Code** Indica el tipus de trama EAP:

  1 – Request

  Enviada per l'*Authenticator* per demanar les credencials d'autenticació al *Supplicant*.

  2 - Response

  Enviada pel *Supplicant* en resposta a un *Request*.

  3 - Success

Enviada per l'*Authenticator* per indicar-li al *Supplicant* que el procés d'autenticació ha finalitzat correctament i que disposa d'accés a la xarxa.

4 - Failure

Enviada per l'*Authenticator* per indicar-li al *Supplicant* que el procés d'autenticació ha fallat i que no disposa d'accés a la xarxa.

- **Identifier** Identificador de la trama. El valor del camp *Identifier* ha de coincidir en les trames *Responses* i els seus corresponents *Requests*.
- **Length** Identifica la llargària de la trama incloent els camps el *Code*, *Identifier*, *Length* i *Data*. Els bytes fora de rang es tractaran com a *padding* i seran ignorats pel receptor. Si el receptor obté un trama EAP on el valor del camp *Length* és més gran que el nombre de bytes rebuts, la trama serà descartada.
- **EAP Data** La informació continguda en aquest camp ve determinada pel tipus de trama EAP. Longitud variable.

2.1.1.3 EAP Methods

Les trames EAP-Method són les que realment contenen la informació relativa a l'autenticació. Es troben situades dins de trames EAP. Existeixen una gran varietat d'EAP-Methods. Alguns es basen en certificats, altres en un nom d'usuari i contrasenya, alguns intercanvien la informació mitjançant un túnel TLS, etc.

Els EAP-Methods s'encarreguen de realitzar el xifrat, desxifrat i empaquetament de la informació d'autenticació.

Els mètodes d'autenticació es poden classificar segons varis criteris:

- **Autenticació simple/mútua**

  En l'autenticació simple només s'autentica el *Supplicant*.

  En l'autenticació mútua s'autentiquen tant el *Supplicant* com l'*Authentication Server*. Aquest tipus d'autenticació evita que el *Supplicant* enviï les seves credencials a un fals servidor.

- **Autenticació mitjançant certificats/clau compartida**

  En l'autenticació mitjançant certificats la verificació de la identitat dels agents involucrats en la comunicació es basa en l'ús de certificats X.509.

En l'autenticació mitjançant clau compartida la verificació de la identitat dels agents involucrats en la comunicació es basa en un nom d'usuari i contrasenya.

- **Autenticació tunelada/no tunelada**

  En l'autenticació tunelada el procés d'autenticació és realitza dins d'un túnel TLS.

  En l'autenticació no tunelada El procés d'autenticació es realitza fora d'un túnel TLS. En aquests casos, les credencials del *Supplicant* acostumen a viatjar xifrades.

La comunicació utilitzant EAP-Methods es realitza entre el *Supplicant* i l'*Authentication Server*. El nombre de missatges a intercanviar dependrà del tipus de EAP-Method utilitzat, que hauran negociat prèviament.

En aquest tipus de trames, l'*Authenticator* juga el paper de mediador: Desencapsula les trames que li arriben del *Supplicant* i les torna a encapsular per a que l'*Authentication Server* les pugui interpretar (i viceversa).

Com s'ha comentat, hi ha una àmplia varietat d'EAP-Methods, alguns d'ells propietaris. Només tres formen part de l'estàndard definit per l'IEEE: *MD5-Challenge*, *One-Time Password* (OTP) i *Generic Token Card* (GTC).

2.1.1.3.1 Trames EAP-Method

A continuació es mostra el format de les trames EAP-Method i es descriuen els seus camps [1]:

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
Code	Length		Data

Figura 2.6 – Format d'una trama EAP-Method

- **Code** Indica l'EAP-Method utilitzat. Cadascun d'ells farà referència a un mètode d'autenticació diferent. A continuació es descriuen les trames especials (aquelles que no són cap mètode d'autenticació):

  1 - Identity

  És utilitzat per transmetre credencials. Aquest tipus de missatges van encapsulats en trames *EAP-Request* i *EAP-Response*.

  2 - Notification

Aquest tipus de trames són utilitzades per l'*Authentication Server* per enviar informació sobre l'estat de l'autenticació (per exemple, pot ser utilitzar per avisar al *Supplicant* que la seva contrasenya expirarà en breu). Alguns EAP-Methods no permeten l'enviament d'aquests tipus de missatges.

3 - NAK (response only)

Aquestes trames són utilitzades pel *Supplicant* per indicar a l'*Authentication Server* que rebutja l'EAP-Method proposat. En la mateixa trama, el *Supplicant* pot suggerir un EAP-Method alternatiu a utilitzar.

254 - Expanded NAK

Aquest tipus de trames també són utilitzades pel *Supplicant* per proposar un EAP-Method alternatiu. La diferència amb les trames NAK és el format del camp *Data*.

- **Length** Indica la longitud del camp *Data*.
- **Data** Conté informació sobre el procés d'autenticació.

2.1.1.3.2 Mètodes d'autenticació

Els mètodes d'autenticació són paquets generats mitjançant algoritmes de xifrat que contenen les credencials de l'usuari (nom d'usuari i contrasenya) i que utilitzarà l'*Authentication Server* per validar-ne la identitat. Hi ha varis mètodes d'autenticació vigents [2]:

- **PAP** (*Password Authentication Protocol*) és el més senzill dels mètodes d'autenticació. Tant el nom d'usuari com la contrasenya es transmeten sense xifrar. El seu ús es recomanat únicament dins d'un túnel TLS.
- **CHAP** (*Challenge Handshake Authentication Protocol*) va ser considerat com una evolució de PAP. En el procés d'autenticació, l'*Authentication Server* envia un *Challenge* al *Supplicant*. El *Supplicant* genera un hash MD5 amb la seva contrasenya i la resposta al *Challenge*. Aquest procés es pot repetir varies vegades.
- **MSCHAPv1** (*Microsoft CHAPv1*) és la primera adaptació del protocol CHAP per a sistemes Microsoft. Aquest mètode d'autenticació ha quedat obsolet en benefici de MSCHAPv2. La principal diferència amb el protocol CHAP és que ni l'*Authentication Server* ni el *Supplicant* emmagatzemen les contrasenyes en text pla. No és necessari emmagatzemar les contrasenyes sense xifrar perquè en el procés d'autenticació s'utilitza només un hash MD5 de les contrasenyes. El nivell de seguretat que aporta aquest mètode d'autenticació no és gaire superior al de CHAP, ja que no és molt difícil obtenir la contrasenya a partir d'un hash MD5.

- **MSCHAPv2** (*Microsoft CHAPv2*) és una versió actualitzada de MSCHAP. Actualment és suportat per tots els sistemes Microsoft. Permet suport per canvi de contrasenyes i missatges de resposta amb estat.

2.1.1.3.3 EAP-Methods

En aquest apartat es descriuen els mètodes d'autenticació basats en EAP:

- **EAP-MD5** (*Message Digest 5*) Protocol d'autenticació simple, no tunelat i basat en clau compartida. És la versió d'EAP més insegura. Mitjançant les credencials d'usuari és genera un hash MD5 (vulnerable a atacs de força bruta). Les dades no viatgen xifrades. Al no generar claus de sessió per al xifrat no és recomanat utilitzar-lo ni en xarxes no cablejades ni en xarxes públiques.

  El seu principal avantatge és la lleugeresa, ja que realitza les operacions de forma ràpida. És fàcil d'implementar i configurar.
- **EAP-OTP** (*One Time Password*) Protocol d'autenticació simple, no tunelat i basat en clau compartida. Té un funcionament idèntic a EAP-MD5, amb la diferència que utilitza el sistema *One Time Password* (sistema generació de claus instantànies) en la resposta.
- **EAP-LEAP** (*Lightweight EAP*) Protocol d'autenticació mútua, no tunelat i basat en clau compartida. És un protocol propietari de Cisco. El funcionament és pràcticament idèntic a l'EAP-MD5. L'única diferència és l'ús d'un sistema de rotació de claus. Aquest EAP-Method s'ha deixat d'utilitzar degut a vulnerabilitats de seguretat descobertes. Ha estat substituït per EAP-FAST.
- **EAP-FAST** *(Flexible Authentication via Secure Tunneling)* Protocol d'autenticació mútua, tunelat i basat en certificats. També desenvolupat per Cisco, pretén solucionar les vulnerabilitats aparegudes en EAP-LEAP. Tunela el transport de l'autenticació mitjançant el sistema anomenat PAC que genera en el servidor una clau única per usuari.
- **EAP-GTC** *(Generic Token Card)* Protocol d'autenticació simple, no tunelat i basat en certificats. Està dissenyat per a l'autenticació d'usuaris mitjançant targetes intel·ligents. És necessari que el *Supplicant* introdueixi el seu *pin* en el procés d'autenticació.
- **EAP-MSCHAP** Protocol d'autenticació simple, no tunelat i basat en claus compartides. És l'aplicació de MSCHAPv1 sobre EAP. És desaconsella el seu ús fora de túnels TLS.

- **EAP-MSCHAPv2** Protocol d'autenticació simple, no tunelat i basat en claus compartides. És l'aplicació de MSCHAPv2 sobre EAP. Tampoc és recomanat el seu ús fora d'un túnel TLS.
- **EAP-SIM** *(Subscriber Identification Module)* Protocol d'autenticació simple, no tunelat i basat en claus compartides. Dissenyat per l'autenticació de dispositius mòbils GSM mitjançant targetes SIM. També es poden autenticar per EAP-SIM altres dispositius sempre que disposin d'un lector de targetes SIM.
- **EAP-AKA** *(Authentication Key Agreement)* Protocol d'autenticació simple, no tunelat i basat en claus compartides. Dissenyat per l'ús en serveis UMTS. Es basa en l'ús de criptografia simètrica per canalitzar l'autenticació i la distribució de claus de sessió. Proporciona privacitat d'usuari i mecanismes de reconnexió ràpida.
- **EAP-TLS** *(Transport Layer Security)* Protocol d'autenticació mútua, tunelat i basat en certificats. El *Supplicant* envia una petició d'autenticació a l'*Authentication Server*, que respon enviant el seu certificat de servidor. Un cop comprovat per part del *Supplicant*, li envia el seu certificat de client, que a la vegada es comprovat pel servidor. Aquest procés es realitza en l'anomenat *outer-tunnel*.

  Un cop s'han verificat les identitats, es crea un túnel TLS per on el *Supplicant* li envia a l'*Authentication Server* les seves credencials i negocien el mètode d'autenticació a utilitzar. Aquest procés es realitza en l'anomenat *inner-tunnel*. L'ús d'aquest túnel TLS ofereix seguretat enfront atacs de diccionari o de MitM.

  Aquest tipus d'autenticació no està lliure d'inconvenients. Degut a la utilització d'autenticació mútua, és necessari que tant el *Supplicant* com el servidor disposin de certificats X.509. Per tant, s'haurà de crear un certificat per a cada usuari al que se li vulgui oferir accés a la xarxa.

  També cal comentar que abans de l'intercanvi de certificats, el *Supplicant* envia a l'*Authentication Server* el seu nom d'usuari en text clar. Un atacant podria obtenir aquesta informació.

  Per últim, les reconnexions són costoses. EAP-TLS genera una gran quantitat de trànsit EAP al realitzar l'autenticació. Cada cop que una sessió EAP-TLS caduca, el procés d'autenticació ha de tornar a començar generant un gran intercanvi de trames.
- **EAP-TTLS** *(Tunneled Transport Layer Security)* és una extensió del protocol EAP-TLS. En aquest mètode només és obligatori que el servidor utilitzi un certificat X.509, tot i que el *Supplicant* també en pot utilitzar.

EAP-TTLS està basat en la creació de dos túnels TLS. El primer s'utilitza per l'intercanvi de credencials, mentre que el segon s'utilitza per l'intercanvi de la clau de xifrat de sessió. Totes les trames EAP viatgen xifrades.

El *Supplicant* comprova el certificat del servidor, mitjançant el qual es crea un túnel en el que s'envien les credencials de l'usuari de forma segura. Gracies a aquest canal de comunicació segur es poden utilitzar altres protocols d'autenticació més simples com PAP o CHAP.

El fet que només el servidor utilitzi certificats simplifica la implementació del mètode, ja que no és necessari crear un certificat per a cada usuari al que se li vulgui donar accés a la xarxa.

EAP-TTLS millora altres vulnerabilitats existents en EAP-TLS. Per exemple, permet que en el primer missatge del procés d'autenticació el *Supplicant* utilitzi un nom anònim com a nom d'usuari, fent que el real no pugui ser interceptat.

EAP-TTLS afegeix l'opció de reconnexió ràpida, permetent continuar la sessió TLS recentment caducada sense generar un gran trànsit de dades.

- **EAP-PEAP** PEAP és un mètode d'autenticació pràcticament idèntic a EAP-TTLS. Va ser desenvolupar per Cisco, Microsoft i RSA, així que és suportat per tots els dispositius d'aquests fabricants.

  Es basa en l'ús de certificats en la banda de l'*Authentication Server* i es compatible amb els mètodes d'autenticació MSCHAPv2 i GTC.

  PEAP i EAP-TTLS són considerats els dos mètodes d'autenticació més segurs, i per tant, els més recomanats.

2.1.1.3.4 Taula comparativa entre els principals EAP-Methods

A continuació es mostra una taula comparativa entre els EAP-Methods més segurs:

	EAP-TLS	EAP-TTLS	EAP-PEAP	EAP-FAST
Certificat de servidor	SI	SI	SI	Shared secret
Certificat client	Obligatori	Opcional	Opcional	PAC
Credencials suportades	Certificats de client	CHAP, PAP, MSCHAPv1, MSCHAPv2	EAP-MSCHAPv2, EAP-GTC, Altres tipus EAP	PAC
Canvi contrasenya	No	Si	Si	Si
Autenticació mútua	Si	Si	Si	Si
Túnel TLS	Si	Si	Si	Si
Entrega de claus dinàmiques	Si	Si	Si	Si
Reconnexió ràpida	Si (a partir RFC5216)	Si	Si	Si
Serveis de directori d'autenticació	AD LDAP	AD LDAP	AD LDAP	AD LDAP
Desenvolupador	Microsoft	Funk, Certicom	Microsoft, Cisco, RSA	Cisco
Suplicants compatibles	Microsoft Open1x Network Manager Wpa_supplicant	SecureW2 Open1x Network Manager Wpa_supplicant	Microsoft SecureW2 Open1x Network Manager Wpa_supplicant	Open1x Wpa_supplicant
Mostra noms usuari	Si	Anònim fase 1	Anònim fase 1	Si
Vulnerable	No	No	No	No
Usos recomanats	802.1x (cablejat i sense fils)	802.1x (cablejat i sense fils)	802.1x (cablejat i sense fils)	Xarxes amb equips Cisco

Taula 2.7 – Comparativa entre els principals mètodes d'autenticació. Font: "RADIUS/AAA/802.1x. Sistemas basados en la autenticación en Windows y GNU/Linux"

2.1.1.4 El procés d'autenticació 802.1x

A continuació es descriu l'intercanvi de missatges entre el *Supplicant* i l'*Authenticator* per dur a terme l'autenticació 802.1x [1]:

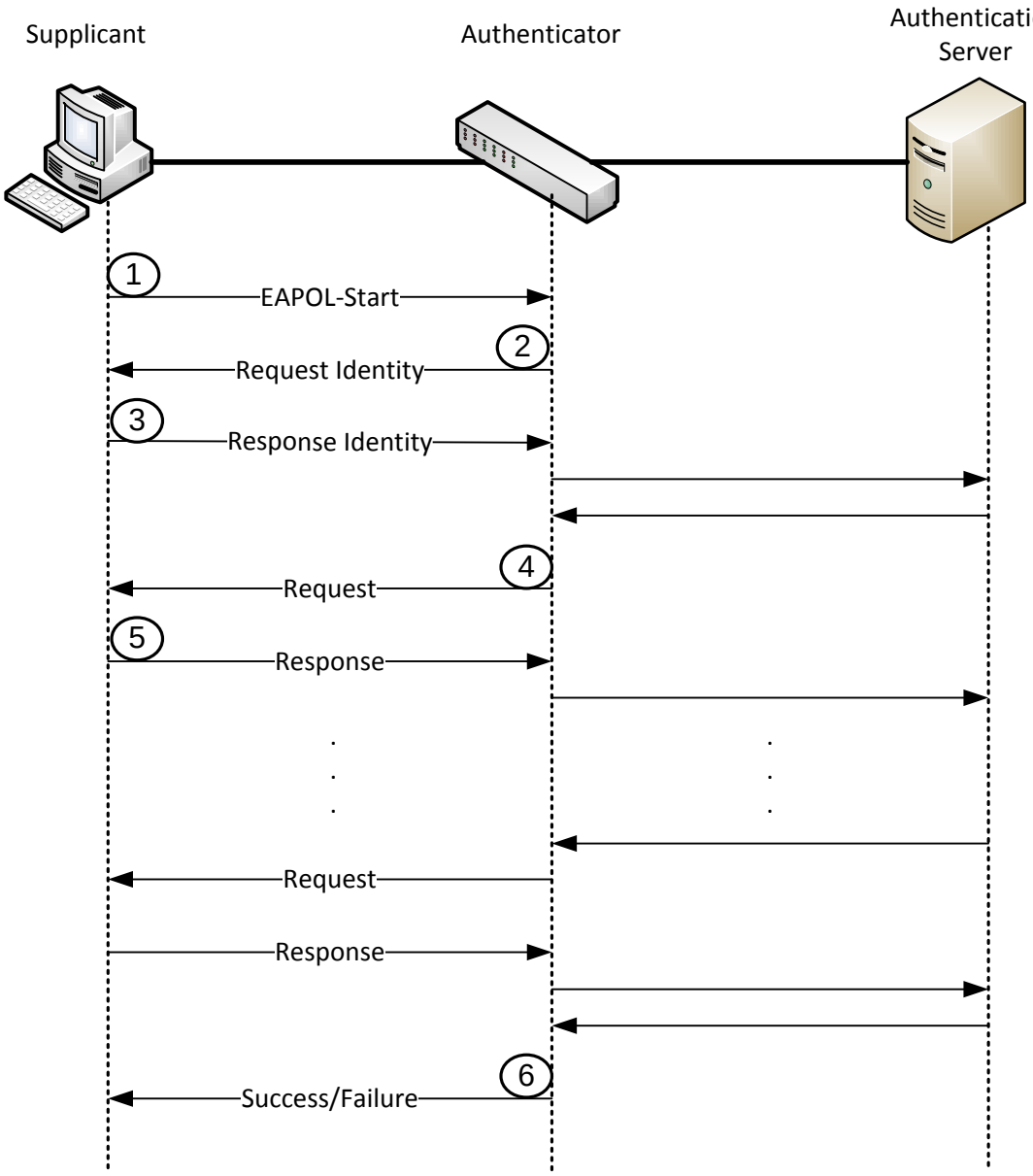


Figura 2.8 – Fase d'autenticació 802.1x

1) El *Supplicant* envia una trama *EAPOL-Start* indicant a l'*Authenticator* que vol iniciar una comunicació EAP.
2) L'*Authenticator* envia un EAP *Request Identity* demanant a l'usuari les seves credencials. L'*Authenticator* enviarà aquest missatge automàticament quan detecti que algun dispositiu s'ha connectat a la interfície. Per tant, en la comunicació 802.1x

l'*Authenticator* és l'agent actiu i no es necessari que hagi rebut un *EAPOL-Start* per enviar un *Request Identity*.

3) El *Supplicant* envia el seu nom d'usuari a l'*Authenticator*, i aquest l'hi reenvia a l'*Authentication Server* utilitzant algun protocol AAA. En aquesta primera trama enviada en text pla, el *Supplicant* no introdueix la seva contrasenya.

   Si l'*Authenticator* no rep un *EAP Response Identity* per part del *Supplicant*, esperarà una estona abans de tornar-li a enviar un nou *EAP Request Identity*. El nombre d'intents i el temps d'espera entre l'enviament de dos *EAP Request Identity* és pot configurar.

   Si finalment el *Supplicant* no respon a les peticions de l'*Authenticator*, es podran prendre una sèrie de mesures. L'acció per defecte és negar-li al *Supplicant* qualsevol tipus d'accés a la xarxa, però també pot ser col·locat en una de les anomenades *Guest VLAN*, on se li permetrà un accés a la xarxa molt restringit.

4) En aquest punt comença un procés de negociació entre l'*Authentication Server* i el *Supplicant* per decidir quin EAP-Method s'utilitzarà en les comunicacions. En aquesta negociació, l'*Authenticator* no prendrà part activa i simplement jugarà el paper d'interlocutor entre tots dos agents.

   L'*Authentication Server* es qui primer proposa un EAP-Method. El *Supplicant* pot acceptar-lo o proposar-ne un de nou (enviant una trama NAK). Aquesta negociació pot constar de varies rondes de missatges depenent de l'EAP-Method utilitzat i dels intents necessaris per arribar a un acord sobre quin EAP-Method a utilitzar.

5) Si finalment no s'arriba a un acord, l'*Authentication Server* li comunicarà a l'*Authenticator* que el procés d'autenticació ha fallat. L'*Authenticator* li comunicarà la decisió al *Supplicant* mitjançant una trama *EAP Failure*. De nou, l'*Authenticator* podrà negar l'accés a la xarxa al *Supplicant* o el podrà col·locar en una *Guest VLAN*.

   En la situació inversa (el *Supplicant* i l'*Authentication Server* es posen d'acord en l'EAP-Method a utilitzar) poden passar dues coses: Que l'autenticació falli (les conseqüències en aquest cas seran les mateixes que les exposades en el cas que no acordin cap EAP-Method) o que el procés d'autenticació sigui satisfactori. En aquest últim cas, l'*Authenticator* enviarà una trama *EAP Success*.

   Si el procés d'autenticació finalitza satisfactòriament, l'*Authentication Server* enviarà a l'*Authenticator* les polítiques d'autorització que s'han d'aplicar al *Supplicant* (per exemple, l'VLAN on ha de ser col·locat o l'*Access List* a aplicar). L'*Authenticator* li comunicarà al *Supplicant* que el procés d'autenticació ha finalitzat correctament mitjançant una trama *EAP Success*.

L'autorització sobre una interfície de l'*Authenticator* és manté inalterable mentre la sessió del *Supplicant* no finalitzi o caduqui. Un cop acabada o caducada, el port tornarà al seu estat inicial.

L'autenticació 802.1x pot tenir caducitat. Aquesta característica es pot configurar en l'*Authenticator*. Quan el temps de vida de la sessió 802.1x finalitza, l'*Authenticator* inicia un nou procés d'autenticació. Molts *Supplicants* permeten realitzar l'autenticació de forma transparent a l'usuari mitjançant les credencials utilitzades en el primer procés d'autenticació.

6) L'*Authenticator* comunica al *Supplicant* el resultat del procés d'autenticació mitjançant un *EAP Success* o un *EAP Failure*.

2.1.1.5 Supplicants 802.1x

Els *Supplicants* són aplicacions software instal·lades en el dispositiu des del qual l'usuari vol obtenir accés a la xarxa. S'encarreguen d'executar la fase d'autenticació comunicant-se amb l'*Authenticator* mitjançant el protocol 802.1x.

En aquest apartat es descriuen i comparen els principals *Supplicants* 802.1x existents tant per sistemes Microsoft Windows com per sistemes Linux.

Les funcionalitats més importants que ha de tenir un *Supplicant* són:

- **Suport per als EAP-Methods més segurs:** És important que un *Supplicant* suporti el màxim nombre d'EAP-methods possible, i especialment aquells que són considerats més segurs. Un *Supplicant* amb poca compatibilitat EAP farà que l'EAP-method a utilitzar en el procés d'autenticació sigui massa dependent del *Supplicant*.
- **Reutilització de credencials:** En molts casos, les credencials utilitzades per un usuari per accedir a la seva sessió de sistema operatiu són les mateixes que les que utilitzarà per autenticar-se a la xarxa. La reutilització de credencials fa referència al fet que el *Supplicant* sigui capaç d'emmagatzemar temporalment les credencials que ha introduït l'usuari per accedir a la sessió del sistema, per a utilitzar-les més tard per autenticar-lo a la xarxa. D'aquesta forma, l'autenticació a la xarxa es realitza de forma transparent a l'usuari, evitant que hagi d'introduir les seves credencials dos cops (una per accedir al sistema operatiu i una altra per obtenir accés a la xarxa).

Aquesta funcionalitat té molta menys importància que la primera, ja que no té res a veure amb la seguretat del sistema. Per altra banda, aporta un grau de comoditat alt per a l'usuari, fent que sigui una funcionalitat que valgui la pena tenir en compte.

2.1.1.5.1 El Supplicant de Microsoft

Els sistemes operatius Windows XP i Windows 7 incorporen aquest *Supplicant* nadiu de Microsoft en la seva instal·lació per defecte.

Permet reutilitzar les credencials d'usuari, però només suporta dos EAP-Methods:

- PEAP
- EAP-TLS

2.1.1.5.2 SecureW2

Un *Supplicant* bastant utilitzat en sistemes Microsoft és l'eina SecureW2 [3]. Les últimes versions d'aquest *Supplicant* són de pagament, tot i que es poden trobar versions anteriors gratuïtes amb pràcticament les mateixes funcionalitats. Suporta més EAP-Methods que el *Supplicant* de Microsoft:

- EAP-TTLS
- PEAP
- EAP-GTC
- EAP-SIM

Per contra, no permet reutilitzar les credencials d'usuari.

2.1.1.5.3 Open1X

Aquest *Supplicant* està disponible per a sistemes Windows XP i Linux [4]. El suport per a Windows 7 es troba en desenvolupament i està previst que s'ofereixi en properes versions.

Suporta una gran varietat d'EAP-Methods:

- PEAP
- EAP-TTLS
- EAP-TLS
- EAP-FAST

- EAP-SIM
- EAP-MD5

Gràcies al mòdul GINA que incorpora, Open1x permet reutilitzar les credencials en sistemes Windows XP. En canvi, no ho permet en sistemes Linux.

2.1.1.5.3 Network Manager

Network Manager és un dels *Supplicants* més populars en sistemes Linux [5]. És compatible amb els següents EAP-Methods:

- EAP-TLS
- EAP-LEAP
- PEAP
- EAP-TTLS

Com la majoria de *Supplicants*, no permet la reutilització de credencials.

2.1.1.5.5 Wpa_supplicant

L'ús de Wpa_supplicant és molt comú en sistemes Linux, tot i que també pot ser utilitzat en sistemes Windows [6]. Suporta l'ús dels principals EAP-Methods:

- EAP-TLS
- EAP-PEAP
- EAP-TTLS
- EAP-SIM
- EAP-FAST
- LEAP

Per defecte, l'Wpa_supplicant no ofereix la possibilitat de reutilitzar les credencials de l'usuari. Però en la seva versió Linux disposa d'una eina anomenada wpa_cli que permet interactuar per línia de comandes amb l'Wpa_supplicant mentre s'està realitzant l'autenticació 802.1x.

Aquesta funcionalitat permet que l'Wpa_cli enviï a l'Wpa_supplicant (en temps d'execució) les credencials de l'usuari sense necessitat que s'hagin de tornar a introduir.

Editant el mòdul PAM de Linux i amb l'ajuda d'*scripts*, es pot aconseguir que l'Wpa_supplicant sigui capaç de reutilitzar credencials.

2.1.1.5.5 Comparativa entre *Supplicants* 802.1x

La següent taula mostra una comparativa entre els diferents *Supplicants* descrits:

	Sistemes operatius			Reutilitza	EAP-Methods							
	Win XP	**Win 7**	**Linux**	**credencials**	**MD5**	**GTC**	**SIM**	**LEAP**	**FAST**	**TLS**	**TTLS**	**PEAP**
Microsoft	SI	SI	NO	SI	NO	NO	NO	NO	NO	SI	NO	SI
SecureW2	SI	SI	NO	NO	NO	SI	SI	NO	NO	NO	SI	SI
Open1x	SI	NO	SI	SI*	SI	NO	SI	NO	SI	SI	SI	SI
Network Manager	NO	NO	SI	NO	NO	NO	NO	SI	NO	SI	SI	SI
Wpa_supplicant	SI	SI	SI	SI**	NO	NO	SI	SI	SI	SI	SI	SI

Taula 2.9 – Comparativa entre els diferents *Supplicants* 802.1x

* Només en Windows XP ** Només en sistemes Linux

2.1.2 VLAN Management Policy Server (VMPS)

VMPS és un protocol de control d'accés basat en ports i desenvolupat per Cisco [7]. Permet assignar VLANs dinàmicament basant-se en l'adreça MAC del dispositiu que està connectat al port. Quan un dispositiu es mogut des d'un port d'un commutador a un altre port d'un altre commutador, el nou commutador assigna al port l'VLAN corresponent per aquell dispositiu.

El servidor VMPS manté un fitxer de text anomenat *address-to-VLAN* on emmagatzema una relació entre les adreces MAC dels dispositius coneguts i l'VLAN associada. El servidor escolta peticions d'autenticació dels seus clients. Quan rep una petició, comprova si l'adreça MAC del dispositiu on està connectat l'usuari es troba en el fitxer *address-to-VLAN*. Si l'adreça MAC coincideix en una de les entrades del fitxer, el servidor acceptarà la petició i li comunicarà al client VMPS la VLAN on s'ha d'assignar l'usuari. En cas que l'adreça MAC no es trobi en el fitxer, l'accés de l'usuari a la xarxa serà denegat.

2.1.2.1 El procés d'autenticació VMPS

A continuació es descriu l'intercanvi de missatges en el procés d'autenticació VMPS:

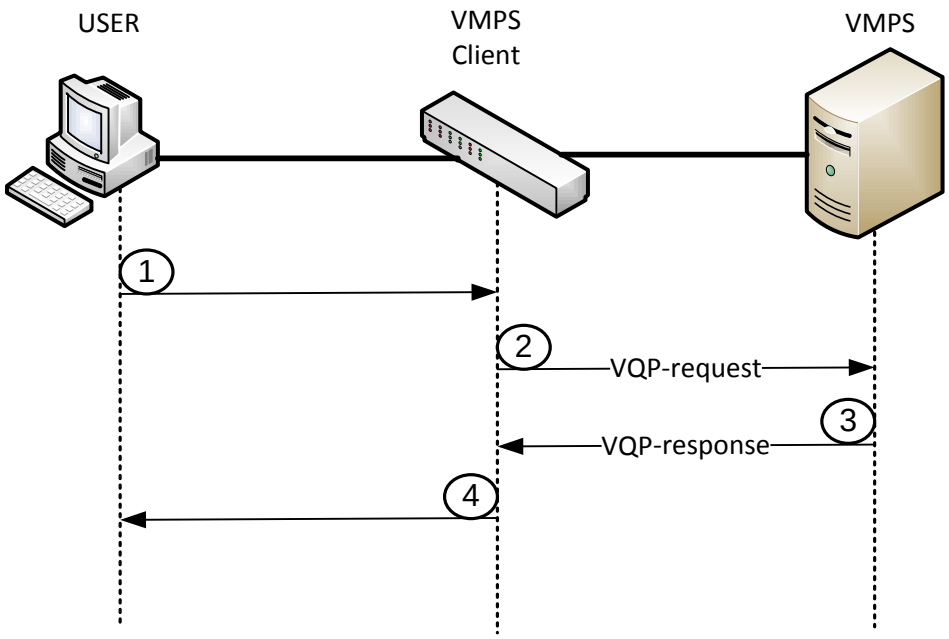


Figura 2.10 – La fase d'autenticació VMPS

1) L'usuari envia trames qualsevols cap al client VMPS.
2) A partir de les trames rebudes, el client VMPS captura l'adreça MAC del dispositiu a través del qual l'usuari vol obtenir accés a la xarxa i li envia al servidor VMPS en una trama *VQP-Request*.
3) El servidor VMPS comprova si l'adreça MAC rebuda té associada alguna VLAN en el seu fitxer *address-to-VLAN*. En cas afirmatiu, el servidor li enviarà al client l'VLAN on s'ha de col·locar l'usuari. En cas contrari, li indicarà al client que no ha de permetre a l'usuari accedir a la xarxa. Aquesta informació se li envia al client en una trama *VQP-Response*.
4) Si la trama *VQP-Response* conté una VLAN, el client VMPS assignarà l'usuari a aquesta VLAN. En cas contrari, el client negarà l'accés de l'usuari a la xarxa.

2.1.3 MAC Authentication Bypass (MAB)

MAC Authentication Bypass és un protocol de control d'accés basat en ports i desenvolupat per Cisco [8]. Es pot entendre com una extensió del protocol VMPS. El seu objectiu també és validar la identitat de l'usuari que vol accedir a la xarxa basant-se en l'adreça MAC del dispositiu des del qual es connecta. MAB està pensat per autenticar aquells dispositius que no suporten el protocol 802.1x.

Molts sistemes operatius porten incorporat el suport per 802.1x, però no tots els possibles *Supplicants* tenen aquesta capacitat. Les impressores, els faxos o els telèfons IP són tres exemples de dispositius que necessiten accés a la xarxa però que no disposen d'un *Supplicant* 802.1x integrat en els seus sistemes operatius.

El funcionament de MAB es basa en les condicions de *timeout* de 802.1x. Quan l'*Authenticator* s'adona que algun dispositiu s'ha connectat a algun dels seus ports, comença el procés d'autenticació 802.1x demanant identificació al *Supplicant*. Mentre l'*Authenticator* no rebi resposta a la seva sol·licitud, seguirà fent peticions per conèixer la identitat del *Supplicant*.

Si al cap d'una sèrie d'intents encara no s'ha rebut resposta per part del *Supplicant*, l'*Authenticator* entendrà que el *Supplicant* no suporta el protocol 802.1x. En aquest punt, descartarà el procés d'autenticació 802.1x i començarà el procés d'autenticació MAB.

Per dur a terme l'autenticació MAB, l'*Authenticator* simplement captura qualsevol trama de les enviades pel *Supplicant* i n'agafa l'adreça MAC. Un cop obtinguda, envia una petició d'autenticació a l'*Authentication Server* (utilitzant un protocol AAA) on els camps dedicats al nom d'usuari i a la contrasenya seran omplerts amb l'adreça MAC del *Supplicant*.

L'*Authentication Server* comprova si l'adreça MAC rebuda és d'un *Supplicant* conegut. En cas afirmatiu, acceptarà la petició d'autenticació i li comunicarà a l'*Authenticator* les polítiques d'autorització que s'han d'aplicar sobre ell.

En cas que l'adreça MAC que li arriba no sigui coneguda, l'*Authentication Server* pot prendre dues decisions: negar l'accés del *Supplicant* a la xarxa o col·locar-lo en una *Guest-VLAN*.

Finalment, l'*Authenticator* li comunica la decisió presa per l'*Authentication Server* al *Supplicant,* i aplica les polítiques d'autorització adients.

2.1.3.1 El procés d'autenticació MAB

A continuació es descriu l'intercanvi de missatges entre el *Supplicant* i l'*Authenticator* per dur a terme l'autenticació MAB:

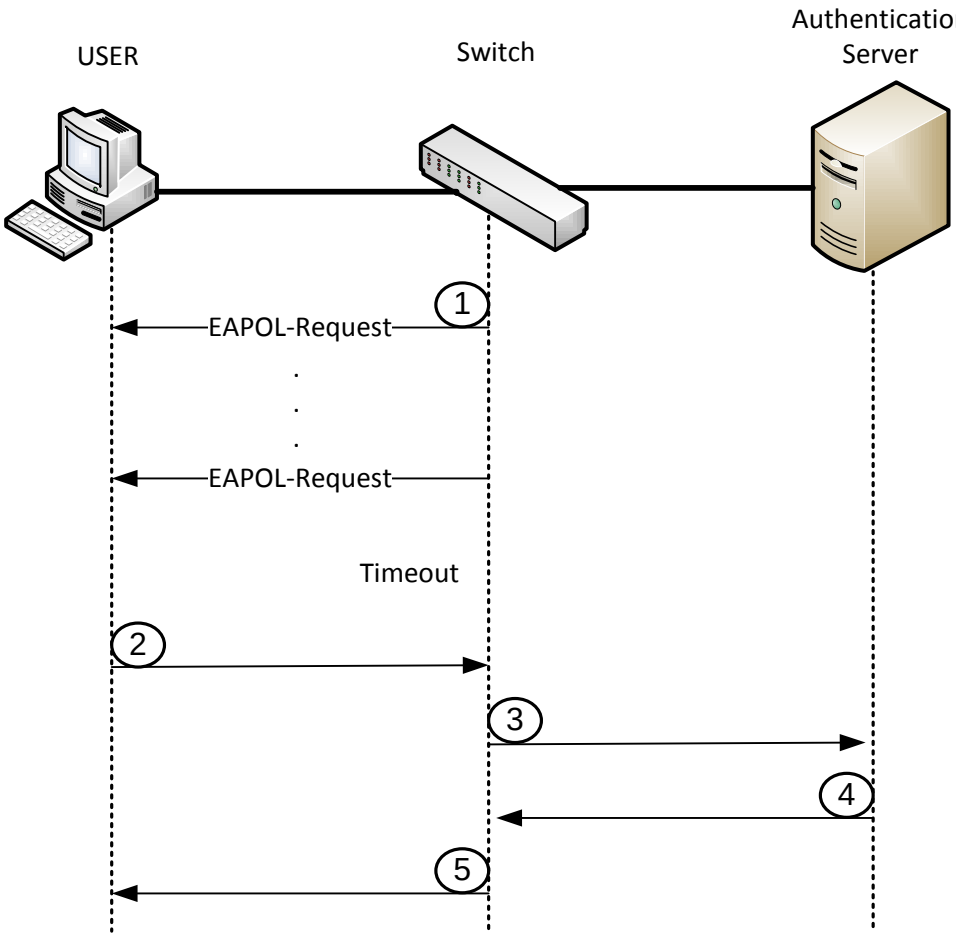


Figura 2.11 – LA fase d'autenticació MAB

1) L'*Authenticator* envia *EAPOL-Request* al *Supplicant* per tal de realitzar l'autenticació 802.1x.
2) Es produeix el *timeout* 802.1x abans que l'*Authenticator* hagi rebut resposta per part del *Supplicant* als *EAPOL-Request* enviats. A partit d'aquí, l'*Authenticator* suposa que l'usuari no disposa d'un *Supplicant* 802.1x i comença el procés d'autenticació MAB. L'*Authenticator* captura l'adreça MAC del *Supplicant*.
3) Un cop coneix l'adreça MAC del *Supplicant*, l'*Authenticator* fa una petició d'autenticació AAA a l'*Authentication Server*.

4) L'*Authentication Server* li comunica a l'*Authenticator* si el procés d'autenticació s'ha dut a terme correctament i quines polítiques d'autorització se li han d'aplicar.
5) L'*Authenticator* li comunica la decisió al *Supplicant* i aplica les polítiques d'autorització corresponents.

La següent figura es resumeix el procés d'autenticació quan es combinen els protocols 802.1x i MAB:

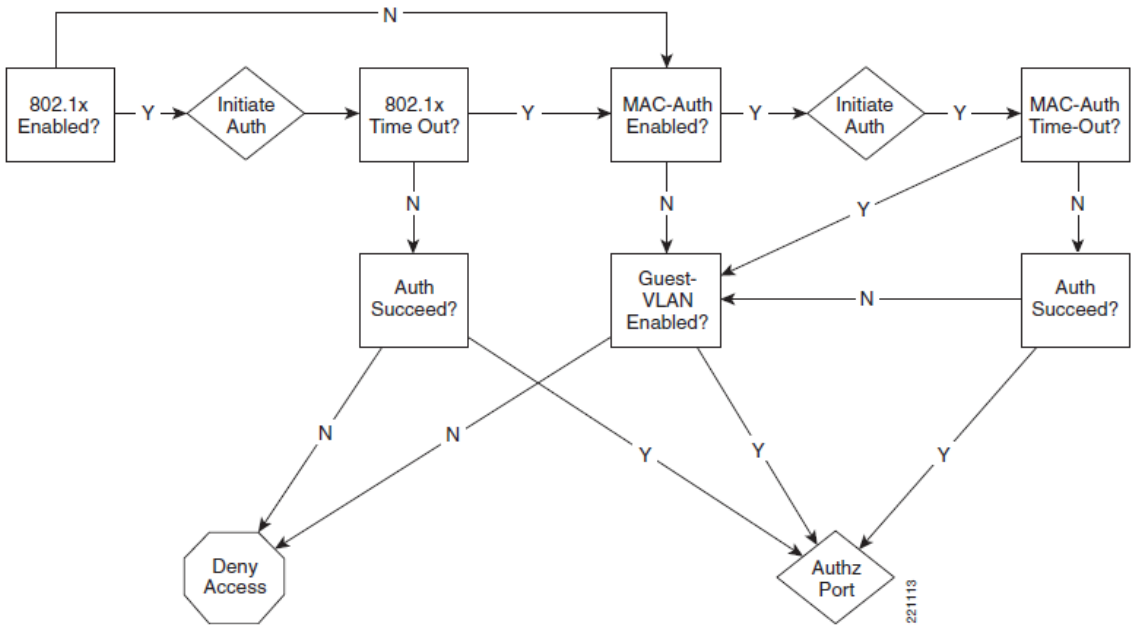


Figura 2.12 – La fase d'autenticació utilitzant conjuntament 802.1x i MAB. Font: www.cisco.com

2.2 AAA

AAA (*Authentication, Authorization and Accounting*) és un *framework* per al disseny de sistemes basats en l'autenticació. Permet controlar qui pot accedir a la xarxa i quins recursos se li atorguen un cop n'ha obtingut l'accés [2] [9]. L'AAA va ser proposat per l'IETF *(Internet Engineering Task Force)* amb l'objectiu de definir un estàndard per a l'autenticació d'usuaris a la xarxa. Aquest estàndard es va basar en el protocol d'autenticació RADIUS, que ja existia per aquell temps.

El *framework* AAA està compost per tres mètodes de seguretat independents entre ells: Autenticació, Autorització i Accounting. Cadascun té com a objectiu respondre una de les següents preguntes:

- **Autenticació** L'usuari és realment que diu ser?
- **Autorització** Quins recursos se li volen oferir a l'usuari?
- ***Accounting*** Com ha utilitzat l'usuari els recursos oferts?

En els següents apartats es descriurà amb més detall cadascuna d'aquestes fases.

A la vegada, els sistemes AAA estan composats per tres agents (amb el mateix rol descrit en l'apartat dedicat al control d'accés basat en ports):

- **Usuari (o *Supplicant*)** És la persona o dispositiu que vol obtenir accés a la xarxa. Envia les seves peticions al client AAA per a que ell les reenviï al servidor. Es comunica amb el client utilitzant protocols basats en ports, com per exemple 802.1x o MAB.
- **Client AAA (o *Authenticator*)** És el dispositiu que fa d'intermediari entre l'usuari i el servidor AAA (normalment un commutador). Les peticions de l'usuari són enviades al servidor utilitzant un protocol AAA. Les respostes del servidor s'envien de nou a l'usuari. És l'encarregat d'aplicar les polítiques d'autorització que li anuncia el servidor.
- **Servidor AAA (o *Authentication Server*)** És l'encarregat de processar les peticions de l'usuari. Gestiona els tres mètodes AAA amb l'ajuda de normes i patrons que té configurats.

La següent figura mostra la distribució dels tres agents a la xarxa:

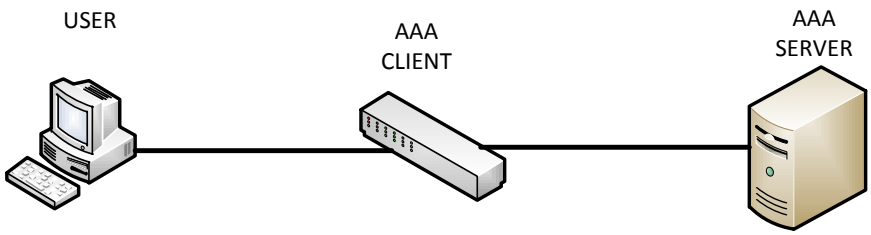


Figura 2.13 – Esquema de xarxa AAA

L'IETF va proposar una sèrie de requisits generals que havia de complir qualsevol protocol AAA. El més significatius eren [10]:

- Suportar milions d'usuaris i milers de connexions simultànies.
- Oferir un servidor secundari en cas de caiguda del servidor principal.
- Proporcionar autenticació mútua entre clients i servidors AAA.
- Oferir confidencialitat de les dades. El client i servidor AAA han de ser els únics agents capaços de poder desxifrar les dades.
- Garantir la integritat de les dades. Cap agent intermedi ha de ser capaç de modificar algun valor dels atributs.
- Oferir suport pel transport de certificats.
- Utilitzar un mecanisme de transport de confiança.
- Suportar IPv4 i IPv6
- Proporcionar suport per *proxy* i control de càrrega.
- Auditar les trames des de que surten del servidor AAA fins que hi tornen.
- Suportar l'ús d'atributs propietaris que no formen part de l'estàndard AAA.

2.2.1 La fase d'autenticació

L'etapa d'autenticació proveeix un mètode per identificar usuaris. En aquesta fase l'usuari facilita unes credencials al servidor AAA i aquest en comprova la seva validesa.

L'usuari es pot identificar de diverses maneres: utilitzant un nom d'usuari i contrasenya, mitjançant un certificat, utilitzant una targeta digital, etc. Per la seva banda, el servidor AAA pot validar les credencials de l'usuari en varies fonts, com per exemple serveis de directori, bases de dades, etc.

De les tres etapes que conjuntament formen el *framework* AAA, la d'autenticació és la més important, ja que si aquesta falla les altres dos no s'arribaran a executar.

Normalment, el servidor AAA disposa d'una sèrie de mètodes d'autenticació disponibles amb un ordre de prioritat. El servidor AAA i el *Supplicant* negociaran el mètode d'autenticació a utilitzar abans d'intercanviar les seves credencials.

L'IETF va definir els requisits que ha de complir un protocol AAA en la seva fase d'autenticació [10]:

- Suportar els mètodes d'autenticació CHAP i PAP.
- Suportar EAP.
- Permetre que l'usuari es pugui re-autenticar sota demanda.
- No demanar les credencials d'usuari fora de la fase d'autenticació.

2.2.3 La fase d'autorització

L'autorització és el procés en el es que decideix quins recursos de xarxa seran oferts a l'usuari. El servidor mantindrà una relació entre cada usuari (o grups d'usuari) i les polítiques d'autorització que se li han d'aplicar.

L'intercanvi d'informació d'autorització entre el servidor i el client AAA es realitza mitjançant AVPs *(Attribute Value Pairs)*, que són definits per cada protocol. Els AVPs simplement són paràmetres amb un valor assignat. El client AAA utilitza els AVP per indicar-li al servidor AAA tota la informació de l'usuari. Per la seva banda, el servidor AAA utilitza els AVP per indicar-li al client AAA les polítiques d'autorització que se li han d'aplicar a l'usuari.

El clients AAA poden tenir els seus propis AVPs anomenats *Vendor Specific Attibute*. Per a poder-los utilitzar en la comunicació és necessari que el servidor AAA ofereixi compatibilitat amb aquests atributs propietaris.

L'IETF va definir els requisits que ha de complir un protocol AAA en la seva fase d'autorització [10]:

- El servidor AAA ha de ser capaç d'assignar adreces IPv4 i IPv6.
- Qualsevol protocol AAA ha de ser compatible amb els AVP definits per RADIUS.
- Els *proxies* poden negar l'accés dels usuaris a la xarxa sense comunicar-li al servidor AAA.

- El servidor AAA pot canviar les polítiques d'autorització d'un usuari en qualsevol moment sense que sigui necessari tornar a realitzar el procés d'autenticació.
- El protocol AAA ha de suportar restriccions d'ús de xarxa com per exemple: *idle timeouts*, filtrat de trames o paràmetres de *QoS*.
- El servidor AAA pot demanar al client AAA una desconnexió de l'usuari en qualsevol moment.

2.2.4 La fase d'*accounting*

L'Accounting és l'última de les etapes AAA i s'executa un cop la fase d'autenticació i autorització han finalitzat.

En aquesta etapa, el client AAA s'encarrega d'obtenir dades estadístiques sobre la sessió d'usuari (temps que ha durat la sessió, nombre de dades transmeses/rebudes, etc.) i enviar-les al servidor AAA per a que les emmagatzemi, normalment en una base de dades.

L'IETF va definir els requisits que ha de complir un protocol AAA en la seva fase d'*Accounting* [10]:

- Oferir *accounting* en temps real.
- No utilitzar grans quantitats de dades, evitant sobrecarregar el sistema.
- Ús d'un *buffer* per emmagatzemar les dades temporalment fins que puguin ser transmeses totes de cop.
- Garantir l'entrega de les dades.
- Utilitzar marques de temps en les dades.
- Poder utilitzar varies entrades en la base de dades per a una única sessió (*accounting* dinàmic).

2.2.5 Protocols AAA

En aquest capítol es descriuen els principals protocols AAA utilitzats en l'actualitat: RADIUS, TACACS+ i Diameter.

Es farà una descripció de cadascun d'ells, explicant els seus trets característics. També es mostrarà l'intercanvi de missatges entre els diferents agents per realitzar les tres fases AAA i el format de les trames utilitzades.

2.2.5.1 RADIUS

RADIUS *(Remote Remote Access Dialin User Service)* és un protocol AAA desenvolupat originalment per Livingston Enterprises. En un principi, l'objectiu de RADIUS era proporcionar accés a la xarxa als usuaris que es connectaven mitjançant mòdems (d'aquí el seu nom).

Amb el temps, RADIUS s'ha convertit en el protocol AAA per excel·lència degut al seu constant creixement i millora. La gran majoria de fabricants han implementat aquest protocol en els seus sistemes propietaris de control d'accés a la xarxa.

A continuació es detallen les principals característiques de RADIUS [11]:

- **Model client/servidor** El client RADIUS és responsable de passar la informació d'usuari als servidors i d'aplicar els paràmetres de connexió que el servidor li comunica. Els servidors RADIUS es troben a la espera de rebre peticions del client. Un cop rebudes, autentiquen i autoritzen l'usuari. Un servidor RADIUS pot actuar com a client *proxy* d'altres servidors RADIUS o altres tipus de servidors d'autenticació.
- **Seguretat de xarxa** Les transaccions entre client i servidor RADIUS són validades utilitzant un *shared secret* que mai es enviat per la xarxa. El *shared secret* s'utilitza per xifrar i desxifrar els camps de les trames destinats a transportar les contrasenyes de l'usuari.
- **Flexibilitat en els mecanismes d'autenticació** El servidors RADIUS suporten una gran varietat de mètodes d'autenticació per dur a terme el procés d'autenticació d'una forma segura. L'objectiu de RADIUS és anar substituint els mètodes que presentin vulnerabilitats pels nous mètodes més segurs que van sorgint.
- **Protocol Extensible** La comunicació entre client i servidor RADIUS es realitza mitjançant AVPs. RADIUS permet la introducció de nous AVPs, convertint-lo en un protocol extensible. Aquest sistema de comunicació mitjançant atributs es un dels principals pilars d'aquest protocol.
- **UDP** RADIUS utilitza UDP per tal de mantenir una còpia de les trames sobre la capa de transport i, així, poder-les reenviar a altres servidors RADIUS en cas d'una fallada en el primer. D'aquesta forma es simplifica el disseny del protocol, evitant fer-se càrrec del control d'arribada d'aquests paquets. Els reenviaments de trames a altres servidors es farà de forma més ràpida, ja que el port no quedarà bloquejat pel control de connexió, com passa en TCP.
- **Multi-plataforma** Es poden trobar servidors RADIUS per la majoria de sistemes: GNU-Linux, Windows, Solaris, ...

- **Repte/resposta** L'autenticació d'usuaris es basa en el procés anomenat repte/resposta. L'usuari que desitja autenticar-se a la xarxa rep un repte per part del servidor. Els usuaris autoritzats estan equipats amb un software que facilita el càlcul de la resposta. Si el servidor rep la resposta esperada seguirà amb el procés d'autenticació. Si no, l'accés de l'usuari a la xarxa serà denegat.

2.2.5.1.1 Autenticació i Autorització en RADIUS

RADIUS no separa els processos d'autenticació i autorització. Si la petició d'autenticació enviada pel client RADIUS és acceptada, el servidor realitzarà en el mateix procés la fase d'autorització i li comunicarà en la resposta al client. A continuació es descriu l'intercanvi de missatges entre els diferents agents implicats per dur a terme la fase d'autenticació/autorització [11]:

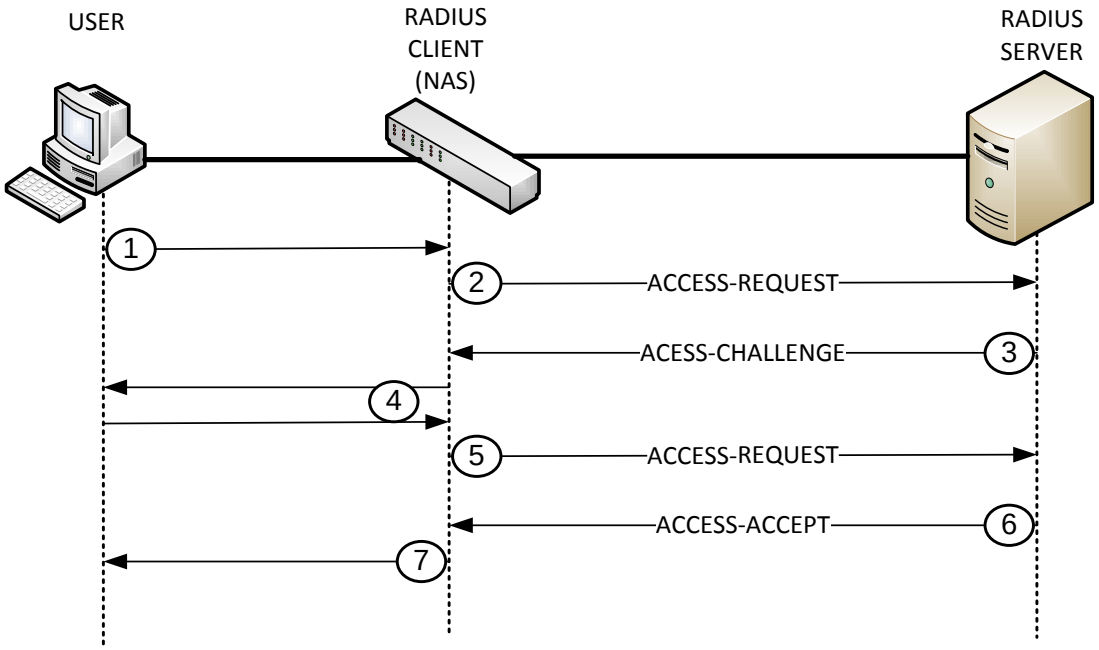


Figura 2..14 – La fase d'autenticació/autorització en RADIUS

1) L'usuari inicia la connexió enviant les seves credencials al client RADIUS.
2) El client RADIUS envia una trama *ACCESS-REQUEST* al servidor utilitzant atributs com el nom d'usuari, la contrasenya, l'identificador de client o l'identificador del port al qual està connectat l'usuari.
3) El servidor valida la identitat del client RADIUS i de l'usuari. Per validar el client es basa en el *shared secret* i per validar l'usuari es basa en les seves credencials. Si algun dels

dos processos de validació no és correcte, el servidor envia un *ACCESS-REJECT* denegant l'accés de l'usuari a la xarxa. Si les dades són correctes, envia un repte (en una trama *ACCESS-CHALLENGE*) que l'usuari haurà de resoldre.

4) El client RADIUS li envia el repte a l'usuari i recull la seva resposta.
5) El client RADIUS envia un nou *ACCESS-REQUEST* al servidor amb la resposta al repte proposat pel servidor. La resposta anirà xifrada i situada en l'atribut dedicat a la contrasenya d'usuari.
6) El servidor pot respondre amb un *ACCESS-ACCEPT* (permetent l'usuari accedir a la xarxa), amb un *ACCEPT-REJECT* (denegant l'accés de l'usuari a la xarxa) o amb un nou *ACCESS-CHALLENGE* (enviant un nou repte a l'usuari). En el cas que enviï un *ACCESS-ACCEPT*, el servidor aprofitarà aquesta trama per indicar-li al client RADIUS les polítiques d'autorització que s'han d'aplicar sobre l'usuari.
7) El client RADIUS notifica a l'usuari la decisió presa pel servidor.

2.2.5.1.2 Accounting en RADIUS

La fase d'*accounting* en RADIUS es divideix en dues parts. La primera es realitza tot just finalitzar la fase d'autenticació/autorització. La segona es realitza un cop la sessió d'usuari ha finalitzat. A continuació es descriu l'intercanvi de missatges entre els diferents agents per dur a terme la fase d'*accounting* [12]:

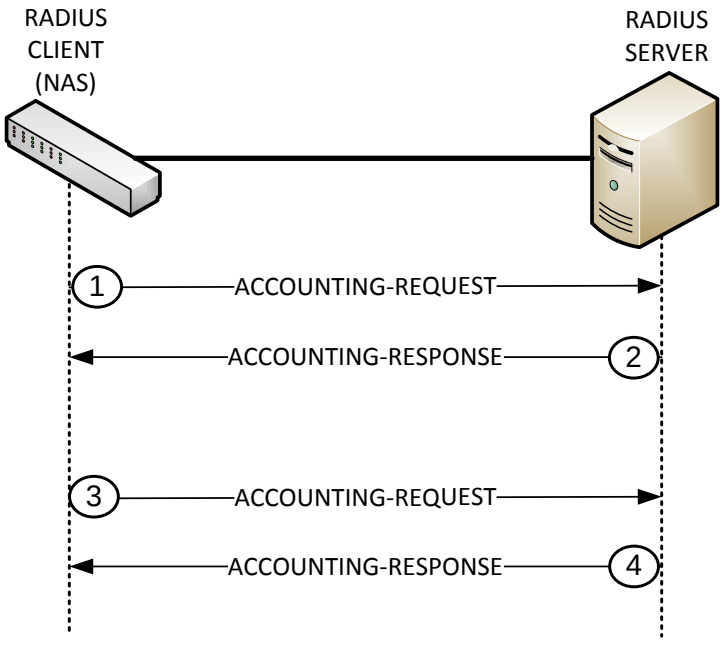


Figura 2.-15 – La fase d'*accounting* en RADIUS

1) El client RADIUS envia una trama *ACCOUNTING-REQUEST* al servidor indicant l'inici de la sessió d'usuari.
2) El servidor RADIUS retorna una trama *ACCOUNTING-RESPONSE* indicant al client que la trama ha estat rebuda.
3) Quan finalitza la sessió, el client RADIUS envia una nova trama *ACCOUNTING-REQUEST* indicant la finalització del servei i enviant les dades estadístiques relacionades amb el procés d'*accounting*.
4) El servidor RADIUS torna a enviar una trama *ACCOUNTING-RESPONSE* indicant al client que la trama ha estat rebuda i les dades emmagatzemades.

2.2.5.1.3 Format de les trames RADIUS

A continuació es descriu el format de les trames RADIUS. Tots els missatges utilitzats en el protocol comparteixen format:

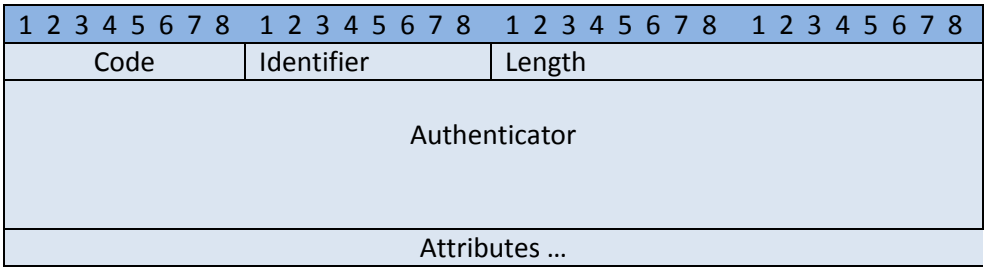


Figura 2.16 – Format d'una trama RADIUS

- **Code** Identifica el tipus de trama:
  - 1 **-** Access-Request
  - 2 - Access-Accept
  - 3 - Access-Reject
  - 4 - Accounting-Request
  - 5 - Accounting-Response
  - 11 - Access-Challenge
  - 12 - Status-Server (experimental)
  - 13 - Status-Client (experimental)
  - 255 - Reserved
- **Identifier** L'objectiu d'aquest camp és relacionar les peticions del client amb les seves corresponents respostes enviades pel servidor. L'identificador serà canviat quan el contingut del camp *Attributes* canviï, o quan ja s'ha rebut resposta a una petició. En la resta de casos el valor del camp *Identifier* no variarà.

- **Length** Indica la llargària de la trama incloent els camps *Code, Identifier, Length, Authenticator* i *Attribute*.
- ***Authenticator*** Aquest camp és utilitzat per autenticar les respostes enviades pel servidor RADIUS i en l'algoritme de xifrat de la contrasenya.

Request Authenticator

En les trames *ACCESS-REQUEST* el valor d'aquest camp és aleatori i s'anomena *Request Authenticator*. Es crea un *hash* MD5 a partir del *shared secret* concatenat amb el *Request Authenticator*. Aquest *hash* generat és sumat (XOR) amb la contrasenya de l'usuari sense xifrar. El resultat de l'operació XOR és el valor que tindrà l'atribut *User-Password* en la trama *ACCESS-REQUEST*.

User-Password Attribute = MD5 {Shared Secret, Request Authenticator} XOR User Password Clear-Text

En les trames *ACCOUNTING-REQUEST* el valor del camp *Request Authenticator* es calcula de la següent manera:

Request Authenticator = MD5{Code, Identifier, Length, 16 bytes de zeros, Requests Attributes, Shared secret}

Response Authenticator

El valor del camp *Authenticator* en les trames *ACCESS-ACCEPT, ACCESS-REJECT, ACESS-CHALLENGE* i *ACCOUNTING-RESPONSE* s'anomena *Response Authenticator* i conté un *hash* MD5 generat a partir de la següent concatenació:

Response Authenticator = MD5{Code, Identifier, Length, Request Authenticator (de la trama que està responent),Response Attributes (si n'hi ha), Shared secret}

2.2.5.2 TACACS+

TACACS+ és fruit de l'evolució dels protocols TACACS i XTATACS [2].

TACACS *(Terminal Access Controller Acess Control System)* va ser un protocol originalment desenvolupat pel Ministeri de Defensa de EEUU i l'empresa BBN Planet Corp. Realitzava autenticacions d'usuaris basant-se en un nom d'usuari i contrasenya, permetent que aquestes credencials poguessin estar emmagatzemades remotament. Era un protocol bastant simple i vulnerable, ja que no utilitzava cap tipus de xifrat sobre les seves trames. Qualsevol persona que interceptés una trama podia tenir accés a les dades d'autenticació d'un usuari.

Igual que RADIUS, es basava en un model client-servidor, on el servidor es mantenia a l'espera de rebre peticions d'autenticació per part del NAS (o client TACACS).

L'any 1990 Cisco va estendre TACACS, afegint-hi la fase d'*accounting*. A aquesta extensió se la va anomenar XTACACS.

Més endavant Cisco va decidir crear TACACS+. Un nou protocol basat en TACACS i XTACACS però bastant diferenciat (fins al punt que era incompatible amb els seus predecessors). TACACS+ compleix els estàndards AAA.

El principal inconvenient de TACACS+ és que és un protocol propietari no obert a altres fabricants. Per aquest motiu, només s'acostuma a utilitzar en infraestructures on tots els dispositius són Cisco.

Les principals característiques de TACACS+ són [13]:

- **Separació de les fases d'autenticació i autorització** Així com RADIUS executa la fases d'autenticació i autorització simultàniament, TACACS+ realitza l'autenticació, l'autorització i l'*accounting* de forma separada.
- **Xifrat de trames** A diferència de RADIUS (que només xifra el camp de contrasenya), TACACS+ xifra completament el cos del missatge, deixant sense xifrar només la capçalera.
- **TCP** TACACS+ utilitza TCP en la capa de transport.
- **Multi-Protocol** TACACS+ suporta els següents protocols: Netbios, X25, AppleTalk i Novell NASI.
- **Gestió de d'encaminadors** TACACS+ proveeix dos mètodes per controlar quines comandes poden utilitzar els usuaris (o grups d'usuaris) en els clients TACACS+ i quines no. El primer mètode consisteix en assignar nivells de privilegis a les comandes. El

client consultarà al servidor TACACS+ si l'usuari en qüestió disposa de suficients privilegis per executar la comanda. El segon mètode consisteix en definir en el servidor TACACS+ la llista d'usuaris (o grups d'usuari) que poden executar cada comanda en el client.

2.2.5.2.1 Autenticació en TACACS+

A continuació es descriu l'intercanvi de trames entre els diferents agents TACACS+ per dur a terme la fase d'autenticació [14]:

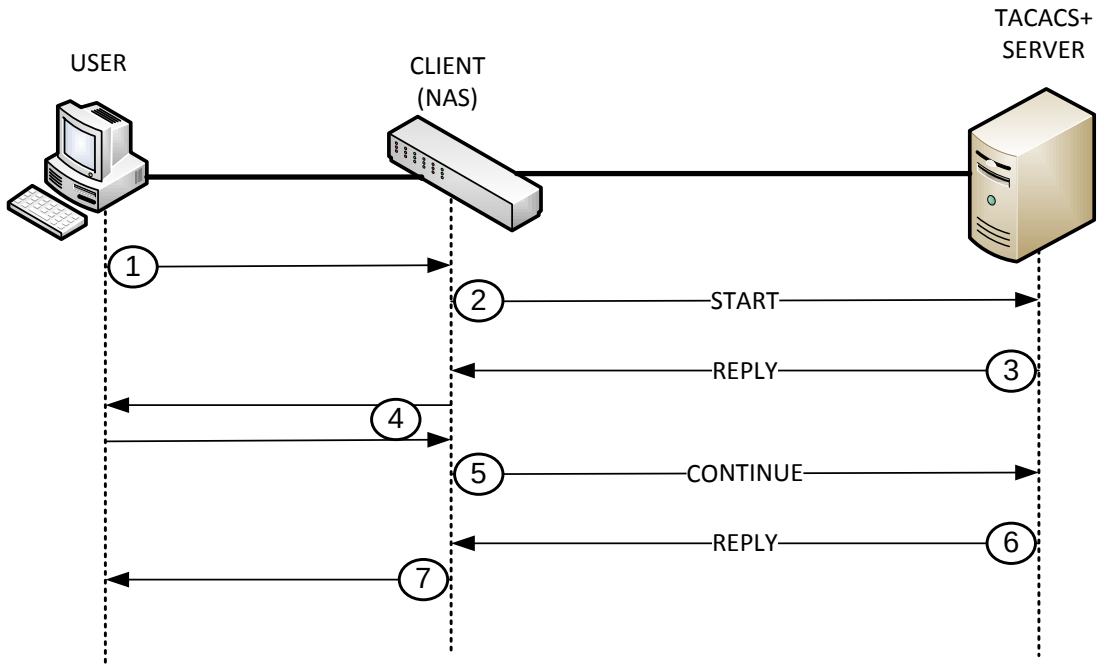


Figura 2.17 – La fase d'autenticació en TACACS+

1) L'usuari inicia la connexió.
2) El client TACACS+ envia una trama *START* al servidor. Les trames *START* descriuen el tipus d'autenticació a utilitzar i poden contenir el nom d'usuari i dades d'autenticació.
3) En resposta a la trama *START*, el servidor envia un *REPLY*. Les trames *REPLY* indiquen si la fase d'autenticació ha acabat o si, per contra, ha de continuar. Si el procés no ha acabat, el servidor indicarà quina nova informació sobre l'usuari vol obtenir.
4) El client TACACS+ demana la nova informació a l'usuari i aquest li facilita.
5) El client TACACS+ envia la nova informació al servidor en una trama *CONTINUE*.
6) Depenent de la informació continguda en la trama *CONTINUE*, el servidor acceptarà o rebutjarà la petició d'autenticació.

7) El client informa de la decisió presa pel servidor TACACS+ a l'usuari.

2.2.5.2.2 Autorització en TACACS+

En la fase d'autorització només es produeix un únic intercanvi de trames petició/resposta entre el client i servidor TACACS+ [14]:

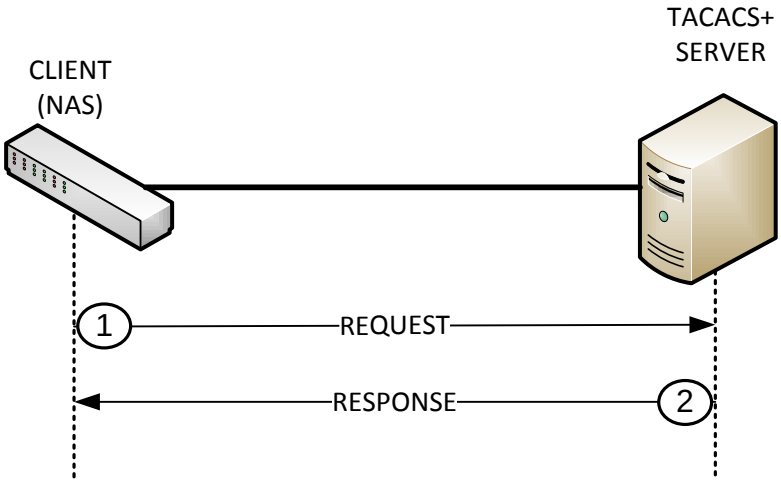


Figura 2.18 – La fase d'autorització en TACACS+

1) El client TACACS+ envia una trama *REQUEST* al servidor indicant els serveis que l'usuari sol·licita.
2) El servidor TACACS+ retorna una trama *RESPONSE* acceptant la petició de l'usuari o rebutjant-la. En aquest últim cas, el servidor indicarà les polítiques d'autorització que s'han d'aplicar.

2.2.5.2.3 *Accounting* en TACACS+

La fase d'*accounting* en TACACS+ és molt similar a la fase d'autorització [14]:

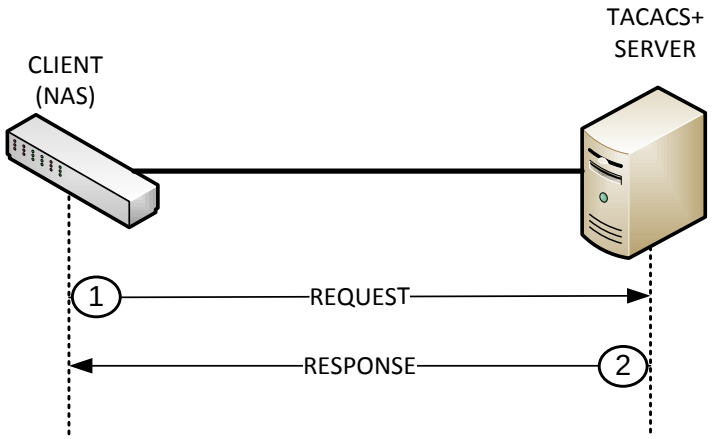


Figura 2.19 - La fase d'*accounting* en TACACS+

1) El client TACACS+ envia una trama *REQUEST* al servidor indicant-li les dades d'*accounting* que han de ser emmagatzemades en la base de dades.
2) El servidor TACACS+ intenta emmagatzemar les dades rebudes i li comunica el resultat al client mitjançant una trama *RESPONSE*.

2.2.5.2.4 Format de les trames TACACS+

En aquest apartat es descriu el format de les trames TACACS+. Al contrari de RADIUS i Diameter, totes les trames TACACS+ tenen un format diferent [14].

2.2.5.2.4.1 Capçalera TACACS+

Totes les trames TACACS+ comencen per la següent capçalera, que sempre s'envia sense xifrar:

1 2 3 4	5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
major version	minor version	type	seq_no	flags
session_id				
length				

Figura 2.20 – Format de la capçalera TACACS+

- **Major version** Número de versió *major* de TACACS+. Identifica el protocol.

  12 - TACACS+ *major version*

- **Minor version** Número de versió *minor* de TACACS+. Cada valor fa referencia a una versió diferent del protocol TACACS+:

  0 - Versió per defecte.

  1 - Versió 1.

- **Type** Indica el tipus de trama TACACS+:

  1 - Autenticació.

  2 - Autorització.

  3 - *Accounting*.

- **Seq_no** Indica el número de seqüència de l'actual trama en l'actual sessió. El *seq_no* de la primera trama TACACS+ valdrà 1 i cada trama augmentarà aquest valor en una unitat. Així, els clients només enviaran trames amb *seq_no* senars i el servidor només enviarà trames amb *seq_no* parells.
- **Flags** Camp utilitzat per definir *flags*:

  tac_plus_unencrypted_flag

  Si s'activa aquest *flag*, la trama no s'envia xifrada. En el cas que aquest bit estigui desactivat la trama és xifrarà utilitzant un *pad*. El *pad* és el resultat d'una sèrie de concatenacions de *hash* MD5 que és generen de la següent forma (on *version* és la combinació dels camps *Major version* i *Minor version*):

  ```
  MD5_1 = MD5{session_id, shared secret, version, seq_no}
  MD5_2 = MD5{session_id, shared secret, version, seq_no, MD5_1}
  ....
  MD5_n = MD5{session_id, shared secret, version, seq_no, MD5_n-1}
  ```

  tac_plus_sigle_connect

  Habilitant aquest *flag*, el client TACACS+ indica que suporta la multiplexació de sessions TACACS+ sobre un única connexió TCP.

  Si el servidor TACACS+ habilita aquest *flag* en la primera trama *REPLY* en resposta a un *REQUEST*, indica la seva voluntat de suportar una connexió simple sobre l'actual connexió.

- **Session_id** Identificador de la sessió TACACS+. El valor d'aquest camp es escollit de forma aleatòria i no pot ser modificat mentre duri la sessió.
- **Length** Longitud de la trama TACACS+ (sense incloure la capçalera).

2.2.5.2.4.2 Trames de la fase d'autenticació

La fase d'autenticació en TACACS+ està composada per tres tipus de trames: *START, CONTINUE* i *REPLY*. Les trames *START* i *CONTINUE* són enviades pel client mentre que les *REPLY* són enviades pel servidor.

START

Les trames *START* són enviades pel client TACACS+ per iniciar el procés d'autenticació:

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
Action	priv_lvl	authen_type	service
user len	port len	rem_addr len	data len
user …			
port …			
rem_addr …			
data …			

Figura 2.21 – Format de la trama START

- **Action** Indica el tipus d'acció que s'ha de dur a terme.
  - 1 - Login.
  - 2 - Canvi de contrasenya.
  - 3 - Enviar contrasenya (obsoleta).
  - 4 - Enviar dades d'autenticació.
- **Priv_lvl** Indica el nivell de privilegi de l'usuari que es vol autenticar. Els diferents nivells de privilegi van des de el 0 fins al 15 en ordre ascendent (com més gran és el valor més gran són els privilegis). Els valors ja definits són:
  - 15 - Nivell de privilegi màxim (*root*).
  - 1 - Nivell de privilegi d'usuari.
  - 0 – Nivell de privilegi mínim.
- **Authen_type** Indica el mètode d'autenticació utilitzat:
  - 1 - ASCII.
  - 2 - PAP.
  - 3 - CHAP.
  - 4 - ARAP.
  - 5 - MSCHAP.

- **Service** Indica el tipus de servei pel qual s'està fent la petició:
  - 0 - *NONE.*
  - 1 - *LOGIN.*
  - 2 - *ENABLE.*
  - 3 - PPP.
  - 4 - ARAP.
  - 5 - PT.
  - 6 - RCMD.
  - 7 - X25.
  - 8 - NASI.
  - 9 - *FWPROXY.*
- **User** Indica el nom d'usuari.
- **Port** Indica el port del client al qual l'usuari està connectat.
- **Rem_addr** Indica la localització remota de l'usuari.
- **Data** Aquest camp és utilitzat per enviar les dades relacionades amb el tipus d'autenticació indicat en el camp *Authen_type*.

REPLY

Les trames *REPLY* són els únics missatges que envia el servidor en la fase d'autenticació. Són la resposta a les trames *START* i *CONTINUE* enviades pel client TACACS+:

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
Status	flags	server_msg len	
data len		server msg …	
data …			

Figura 2.22 – Format de la trama REPLY en la fase d'autenticació

- **Status** Indica l'estat en el qual es troba el procés d'autenticació.
  - 1 - Autenticació acceptada.
  - 2 - Autenticació fallida.
  - 3 - El servidor sol·licita més informació.
  - 4 - El servidor sol·licita el nom d'usuari.
  - 5 - El servidor sol·licita la contrasenya.
  - 6 - Reiniciar.

7 - Error.

8 - L'autenticació s'ha de realitzar en un altre servidor.

- **Flags** Defineixen l'acció a executar.
- **server_msg** Indica el missatge que se li ha de mostrar a l'usuari.
- **Data** Conté les dades relacionades amb les credencials de la fase d'autenticació.

CONTINUE

Les trames *CONTIUNE* són enviades pel client TACACS+ com a resposta a un *REPLY* enviat pel servidor on li sol·licitava al client més informació sobre l'usuari.

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
User_msg len		data len	
flags	User_msg …		
Data …			

Figura 2.23 – Format de la trama CONTINUE

- **Flags** Permeten definir l'acció a executar.
- **user_msg** Conté la resposta de l'usuari al missatge que se li ha mostrat i que anava situat en el camp *Server_msg* de la trama *REPLY* anterior.
- **Data** Conté informació relacionada amb les credencials de l'usuari que havia sol·licitat el servidor per realitzar la fase d'autenticació.

2.2.5.2.4.3 Trames de la fase d'autorització

Dos tipus de trames diferents s'intercanvien el client i servidor TACACS+ en la fase d'autorització: *REQUESTS* i *RESPONSES*

REQUEST

Les trames *REQUESTS* són enviades pel client TACACS+. En elles es defineixen (mitjançant AVPs) les polítiques d'autorització que sol·licita l'usuari.

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
Authen_method	Priv_lvl	Authen_type	Authen_service
User len	Port len	Rem_addr len	Arg_cnt
Arg 1 len	Arg 2 len	…	Arg N len
User …			
Port …			
Rem_addr …			
Arg1 …			
Arg 2 …			
…			
Arg N …			

Figura 2.24 – Format de la trama REQUEST en la fase d'autorització

- **Authen_method** Indica el mètode d'autenticació utilitzat pel client TACACS+ per aconseguir la informació d'usuari:
  - 0 - No assignat.
  - 1 - Cap.
  - 2 - Kerberos 5.
  - 3 - *Line*.
  - 4 - Autenticació per obtenir nous privilegis.
  - 5 - Base de dades local.
  - 6 - Protocol d'autenticació TACACS+.
  - 8 - Autenticació d'invitat.
  - 16 - Protocol d'autenticació RADIUS.
  - 17 - Kerberos 4.
  - 32 - Autenticació mitjançant *R-Commands* de Berkeley Unix.
- **Arg_cnt** Nombre d'AVPs que s'inclouen en la trama.
- **User** Indica el nom d'usuari.
- **Arg** Llista d'AVPs que contenen les polítiques d'autorització que sol·licita l'usuari.
- **Priv_lvl, Authen_type, Authen_service, Port, Rem_addr** tenen la mateixa definició que la descrita en les trames *START* de la fase d'autenticació.

RESPONSE

Les trames *RESPONSE* són enviades pel servidor TACACS+ en resposta a una trama *REQUEST*:

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
Status	Arg_cnt	Server_msg len	
Data len		Arg 1 len	Arg 2 len
…	Arg N len	Server_msg …	
Data …			
Arg 1 …			
Arg 2 …			
…			
Arg N			

Figura 2.25 – Format de la trama RESPONSE

- **Status** Indica l'estat en el que es troba la fase d'autorització:

  1 - Les peticions de l'usuari han estat acceptades.

  2 - La petició de l'usuari serà reemplaçada per les polítiques definides pel servidor en els AVPs de la trama enviada.

  16 - La petició d'autorització ha estat denegada.

  17 - Error en el procés d'autorització.

  33 - La petició d'autorització s'ha de realitzar en un altre servidor.
- **Server_msg** Missatge del servidor que pot ser mostrat a l'usuari si el client ho desitja.
- **Data** Missatge administratiu que pot ser mostrat en una consola o com a *log*. La decisió de mostrar aquest missatge també és del client.
- **arg_cnt** Nombre d'AVPs que conté la trama *RESPONSE*.

2.2.5.2.4.4 Trames de la fase d'Accounting

Igual que en la fase d'autorització, en la fase d'*Accounting* s'utilitzen trames *REQUEST* i *RESPONSE*.

REQUEST

Les trames *REQUEST* són enviades pel client TACACS+ al servidor i contenen les dades d'*accounting* sobre la sessió de l'usuari que han de ser emmagatzemades en la base de dades.

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
flags	Authen_method	Priv_lvl	Authen_type
Authen_service	User len	Port len	Rem_addr len
Arg_cnt	Arg 1 len	Arg 2 len	…
Arg N len	User…		
Port …			
Rem_addr …			
Arg 1 …			
Arg 2 …			
…			
Arg N			

Figura 2.26 – Format de la trama REQUEST en la fase d'*accounting*

Tots els camps ja han estat definits prèviament en la descripció de les fases d'autenticació i autorització.

REPLY

Aquesta trama es enviada pel servidor TACACS+ i es utilitzada per confirmar al client que les dades d'*accounting* enviades han estat emmagatzemades correctament.

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
Server_msg len		Data len	
status	Server_msg		
Data...			

Figura 2.27 – Format de la trama REPLY en la fase d'*accounting*

- **Status** Indica l'estat en el que es troba la fase d'Accounting

  1 - Les dades han estat emmagatzemades correctament.

  2 - Error.

  33 - L'accounting s'ha de realitzar en un altre servidor.

Els camps **Server_msg** i **Data** ja han estat descrits en les trames *REQUEST* de la fase d'autorització.

2.2.5.3 Diameter

Diameter va sorgir amb l'objectiu de ser el protocol AAA successor de RADIUS [2]. Els seus creadors el va definir com "dos cops tant bo com RADIUS" (d'aquí el seu nom). Diameter va ser dissenyat l'any 1996 per la companyia Black Storm Network.

Per facilitar la migració de sistemes basats en RADIUS, es va decidir que Diameter fos compatible amb ell. Per aquest motiu, cada cop que sorgeix una modificació en els RFC de RADIUS, Diameter també ha de ser actualitzat.

Diameter volia millorar totes les limitacions que oferia RADIUS per aquella època. Però RADIUS, amb el temps, ha anat evolucionant i reduint les seves carències fent que Diameter no aportés grans millores sobre ell. Per aquest motiu, l'ús de Diameter s'ha anat deixant de banda en benefici de RADIUS, que s'ha consolidat com el gran estàndard AAA.

Les principals característiques de Diameter són [15]:

- **Transport** Diameter utilitza TCP i SCTP en la capa de transport per gestionar les comunicacions.
- **Failover** Diameter suporta l'enviament de missatges per part del servidor indicant al client que no estarà disponible durant un període de temps i que haurà d'enviar les peticions a un servidor secundari.
- **End-To-End** Mentre que RADIUS únicament suporta la seguretat *Hop-By-Hop*, Diameter també suporta seguretat *End-To-End*. D'aquesta forma cap agent intermedi pot modificar un missatge sense que se'n tingui constància.
- **Missatges d'error** En RADIUS, quan el servidor rep una trama errònia simplement la descarta. En Diameter, el servidor notifica al client qualsevol incidència enviant-li missatges d'error.
- **Longitud de les trames** RADIUS només reserva 1 byte al camp *Length* fent que la llargada màxima de les trames sigui de 255 bytes. Diameter dedica 3 bytes a aquest camp, permetent que la longitud dels missatges pugui ser de 16535 bytes.

2.2.5.3.1 Autenticació/Autorització en Diameter

Diameter utilitza el mateix tipus de trama per dur a terme les fases d'autenticació i autorització.

A continuació es descriu l'intercanvi de trames Diameter per dur a terme la fase d'autenticació/autorització [15]:

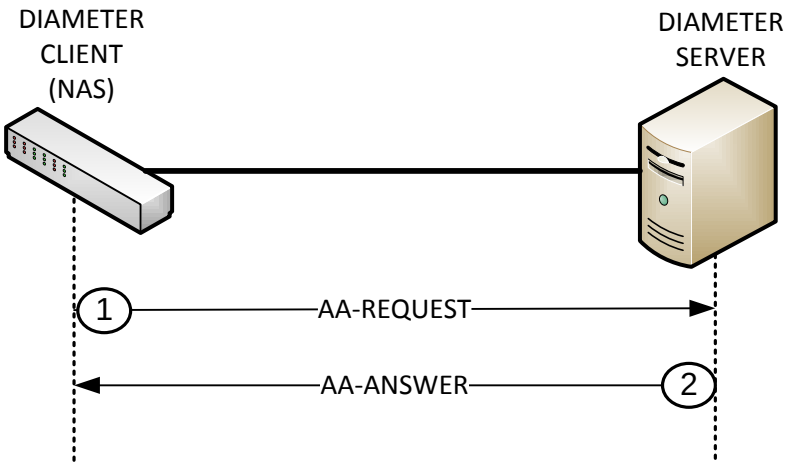


Figura 2.28 – La fase d'autenticació/autorització en Diameter

1) El client Diameter envia una trama *AA-Request* al servidor per autenticar i/o autoritzar l'usuari.
2) El servidor Diameter respon al client amb una trama *AA-Answer* indicant el resultat de l'operació d'autenticació/autorització.

Aquest procés es pot dur a terme mitjançant varis intercanvis de trames *AA-Request* i *AA-Answer*.

2.2.5.3.2 Accounting en Diameter

En la fase d'*accounting*, Diameter també utilitza únicament dos tipus de trames: *ACCOUNTING-REQUEST* (enviades pel client Diameter) i *ACCOUNTING-ANSWER* (enviades pel servidor Diameter).

A continuació es descriu l'intercanvi de trames Diameter per dur a terme la fase d'*accounting* [15]:

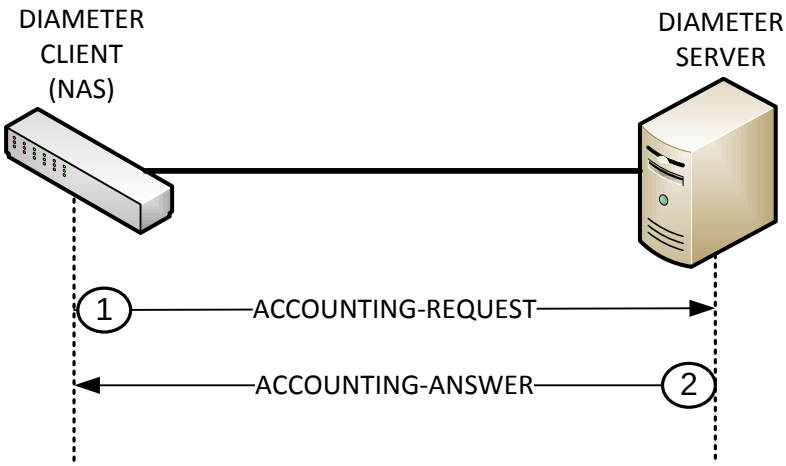


Figura 2.29 – La fase d'*accounting* en Diameter

1) El client Diameter envia una trama *ACCOUNTING-REQUEST* al servidor amb tota la informació sobre la sessió d'usuari que ha de ser emmagatzemada.
2) El servidor Diameter realitza l'operació i li comunica el resultat al client mitjançant una trama *ACCOUNTING-ANSWER*.

2.2.5.3.3 Format de les trames Diameter

El format de totes les trames Diameter és el mateix. Estan formades per una capçalera i una sèrie d'AVPs. Per identificar el tipus de trama enviada s'utilitza el camp *Application-Id* que indica la fase a la que pertany la trama [15].

2.2.5.3.3.1 Capçalera Diameter

Totes les trames Diameter comparteixen la següent capçalera:

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
Version	Message length		
Command flags	Command-code		
Application-ID			
Hop-By-Hop Identifier			
End-To-End Identifier			
AVPs…			

Figura 2.30 – Capçalera de les trames Diameter

- **Version** Indica la versió del protocol. L'actual és la 1.
- **Message Length** Indica la llargària del missatge incloent la capçalera.
- **Command Flags**

1	2	3	4	5	6	7	8
R	P	E	T	r	r	r	r

R(equest)

Si està habilitat, la trama és una petició. Si no, és una resposta.

P (roxiable):

Si està habilitat, la trama ha de ser redirigida. Si no, la trama ha de ser tractada localment.

E(rror):

Si està habilitat, la trama conté un error.

T(Potentially re-transmitted message):

Aquest *flag* s'habilita quan es reenvia una trama la qual no s'ha rebut confirmació de recepció. Indica que la trama podria ser un duplicat. Quan s'envia una trama per primer cop, aquest bit ha d'estar desactivat.

r(eserved):

Aquets bits estan reservats per a un ús futur i han de valer 0. Seran ignorats pel receptor.

- **Command-code** Indica la comanda associada a la trama.
- **Application-Id** Indica el tipus de trama. Aquest camp pot indicar una trama d'autenticació/autorització o d'*accounting*.
- **Hop-by-Hop Identifier** Aquest camp conté un valor enter que té com a objectiu relacionar les peticions amb les seves respectives respostes. L'agent que envia la petició s'ha d'assegurar que el valor assignat a aquest camp no s'hagi utilitzat abans. Normalment en la primera trama s'escull un valor aleatori i cada cop que s'envia una nova trama aquest valor s'incrementa en una unitat.
- **End-To-End Identifier** Aquest camp s'utilitza per detectar missatges duplicats. L'agent que envia la trama assigna un identificador únic a cada missatge. Durant 4 minuts l'identificador no pot ser repetit. El receptor de la trama comprovarà l'identificador del transmissor i el valor d'aquest camp per detectar trames duplicades.

- **AVP** En aquest camp s'envien les dades rellevants de les trames Diameter. El seu format es descriu en el següent apartat.

2.2.5.3.3.2 Capçaleres AVP

Els AVPs de Diameter contenen la informació relacionada amb l'autenticació, l'autorització i l'*accounting*. El seu format és el següent:

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
AVP Code			
V M P r r r r r	AVP Length		
Vendor-ID (opt)			
Data …			

Figura 2.31 – Capçalera AVP de les trames Diameter

- **AVP Code** La combinació d'aquest camp amb el *Vendor-Id-Field* identifica un atribut únic. Els AVP amb identificador entre 1 i 255 estan reservats per oferir compatibilitat amb RADIUS i en aquest cas no és necessari assignar un valor al camp *Vendor-Id*. Els valor AVP a partir del 256 són exclusius de Diameter i estan definits per la IANA.
- **AVP Flags**

  V(endor-ID)

  Indica si s'utilitza el camp *Vendor-ID.*

  P

  Indica si es necessita xifrat *end-to-end*.

  M(andatory)

  Indica si és obligatori que el receptor de la trama suporti AVPs. Si aquest bit està habilitat significa que la informació continguda en el camp *Data* és important. Si aquest bit no està habilitat, significa que la trama és informativa i el receptor pot ignorar l'AVP.

  r(eserved)

  Aquests bits estan reservats per a usos futurs i han de valdre 0.

- **AVP Length** Indica la longitud de la trama AVP incloent els camps *AVP Code, AVP Length, AVP Flags, Vendor-Id* i *Data*.

- **Vendor-ID** Conté l'identificador de l'AVP.
- **Data** Conté el valor de l'AVP. El format i la grandària del valor ve determinat pels camps *AVP Code* i *AVP Length*.

2.2.6 Servidors AAA

Existeixen una gran varietat de servidors AAA. Descriure'l tots provocaria una extensió considerable del document. Per aquest motiu s'ha decidit resumir aquesta descripció mitjançant dues taules comparatives dels servidors AAA més utilitzats tant per RADIUS, TACACS+ i Diameter.

En la primera taula es comparen les característiques i funcionalitats dels diferents servidors. En la segona es mostren els mètodes d'autenticació suportats per cadascun d'ells.

Servidor	Sistemes operatius		Protocols AAA			Bases de dades				Serveis de directori		Software lliure?
	UNIX	MICROSOFT	RADIUS	TACACS+	Diameter	MySQL	Oracle	PostgreSQL	SQL Server	LDAP	Active Directory	
FreeRADIUS [16]	SI	NO	SI	NO	NO	SI	SI	SI	SI	SI	SI	SI
CISCO ACS [17]	NO	SI	SI	SI	NO	NO	SI	NO	SI	SI	SI	NO
MICROSOFT IAS [18]	NO	SI	SI	NO	NO	NO	NO	NO	SI	NO	SI	NO
JUNIPER SBR [19]	SI	SI	SI	NO	NO	SI	SI	NO	NO	SI	SI	NO
GNU RADIUS [20]	SI	NO	SI	NO	NO	SI	NO	SI	NO	NO	NO	SI
OpenRADIUS [21]	SI	NO	SI	NO	NO	SI	NO	NO	NO	SI	NO	SI
BSDRadius [22]	SI	NO	SI	NO	NO	SI	SI	SI	NO	NO	NO	SI
TekRADIUS [23]	NO	SI	SI	NO	NO	NO	NO	NO	SI	NO	SI	SI
OpenDiameter [24]	SI	SI	NO	NO	SI	NO	NO	NO	NO	NO	NO	SI
FreeDiameter [25]	SI	NO	NO	NO	SI	SI	NO	SI	NO	NO	NO	SI

Taula 2.32 – Comparativa (I) entre els principals servidor AAA

Servidor	Mètodes d'autenticació											
	PAP	CHAP	MSCHAP	MSCHAPv2	EAP-GTC	EAP-MD5	EAP-SIM	EAP-LEAP	EAP-FAST	EAP-TLS	EAP-TTLS	EAP-PEAP
FreeRADIUS [16]	SI	SI	SI	SI	NO	SI	SI	SI	NO	SI	SI	SI
CISCO ACS [17]	SI	SI	SI	NO	SI	SI	NO	SI	SI	SI	NO	SI
MICROSOFT IAS [18]	SI	SI	SI	SI	NO	SI	NO	NO	NO	SI	NO	SI
JUNIPER SBR [19]	SI	SI	SI	SI	NO	SI	NO	SI	SI	SI	SI	SI
GNU RADIUS [20]	SI	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
OpenRADIUS [21]	SI	SI	NO	NO	NO	NO	NO	NO	NO	SI	NO	NO
BSDRadius [22]	NO	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
TekRADIUS [23]	SI	SI	SI	SI	NO	SI	NO	NO	NO	SI	NO	SI
OpenDiameter [24]	NO	NO	NO	NO	NO	SI	NO	NO	NO	SI	NO	NO
FreeDiameter [25]	NO	NO	NO	NO	NO	SI	NO	NO	NO	SI	NO	NO

Taula 2.33 – Comparativa (II) entre els principals servidors AAA

2.3 Un sistema real de control d'accés a la xarxa: Eduroam

Abans de finalitzar l'etapa de recerca i començar la de disseny, es va fer un estudi d'un cas real de control d'accés a la xarxa per tal de veure el seu disseny i tecnologies utilitzades. El sistema utilitzat per dur a terme aquest estudi va ser Eduroam [26], el mètode utilitzat per l'UPC per donar accés als usuaris a la xarxa sense fils. L'elecció d'Eduroam va ser deguda a la proximitat amb el sistema motiu d'aquest projecte. Mentre un permet als usuaris de la FIB accedir a la xarxa sense fils (Eduroam), l'altre els permetrà accedir a la xarxa cablejada.

Com s'explicarà més endavant, la gran diferència entre els dos sistemes radica en la infraestructura. Eduroam és un sistema a nivell mundial amb un gran nombre de servidors i *proxies*.

A continuació es farà una breu introducció a Eduroam i s'explicarà el seu funcionament.

2.3.1 Introducció a Eduroam

Eduroam *(EDUcation ROAming)* és el servei mundial de mobilitat segura desenvolupat per la comunitat acadèmica i de recerca. Eduroam facilita la mobilitat dels investigadors i els estudiants europeus, ja que els ofereix connectivitat sense fils en els seus desplaçaments a la resta d'institucions que estan connectades. D'aquesta manera, els usuaris de les institucions que participen a Eduroam tenen accés a Internet a través de les xarxes sense fils de la resta d'institucions participants.

Eduroam ES és una iniciativa englobada en el projecte redIris i que s'encarrega de coordinar a nivell nacional els esforços d'institucions acadèmiques amb la finalitat d'obtenir un espai únic de mobilitat . La Universitat Politècnica de Catalunya participa en el projecte Eduroam, donant accés als seus propis usuaris i als usuaris d'altres institucions participants en Eduroam, mitjançant la coordinació del CESCA (Centre de Supercomputació de Catalunya), que realitza l'enllaç tècnic i administratiu entre les diferents institucions participants en el projecte Eduroam.

2.3.2 Funcionament d'Eduroam

Eduroam realitza les fases d'autenticació i autorització d'usuaris:

- **Autenticació** Es realitza en els anomenats IdP *(Identity Providers)*.
- **Autorització** Es realitza en els SP *(Service Providers)*.

Per a transportar les peticions d'autenticació d'un usuari des del SP fins al seu IdP, es va crear una jerarquia de servidors RADIUS. Cada IdP gestiona el seu propi servidor RADIUS, el qual es connectat a una base de dades local. Aquests servidors RADIUS estan connectats a un servidor RADIUS nacional centralitzat, que a la vegada està connectat als servidors RADIUS de primer nivell (TLR). Tot seguit es descriuen els agents que formen part de la infraestructura Eduroam:

- **TLR** *(Confederation Top-Level RADIUS Servers)* Són els servidors RADIUS de primer nivell (Europeus/Mundials). Accepten les peticions dels FLR dels quals en són responsables, i les reenvien als TLRs responsables del FLR als quals pertany l'usuari que s'ha d'autenticar. Per tant, només actuen com a *proxy* RADIUS.
- **FLR** *(Federation-Level RADIUS Servers)* Reben peticions dels IdP i SP dels quals en són responsables i les reenvien a un altre IdP/SP (si en són responsables) o al seu TLR associat. Els FLR també actuen només com a *proxy* RADIUS.
- **IdP/SP** Els IdP són servidors RADIUS responsables d'autenticar els seus propis usuaris. Els SP són els punts d'accés i commutadors del sistema. Els SP normalment tenen associat un servidor RADIUS que reenvia les peticions dels usuaris externs als FLR. Els SP també s'encarreguen de realitzar la fase d'autorització assignant l'usuari a una VLAN un cop s'ha autenticat. Normalment, les institucions que formen part d'Eduroam acostumen a ser IdP i SP al mateix temps, gràcies a un servidor RADIUS que pot actuar de servidor i de *proxy*.

La següent figura resumeix la jerarquia d'Eduroam:

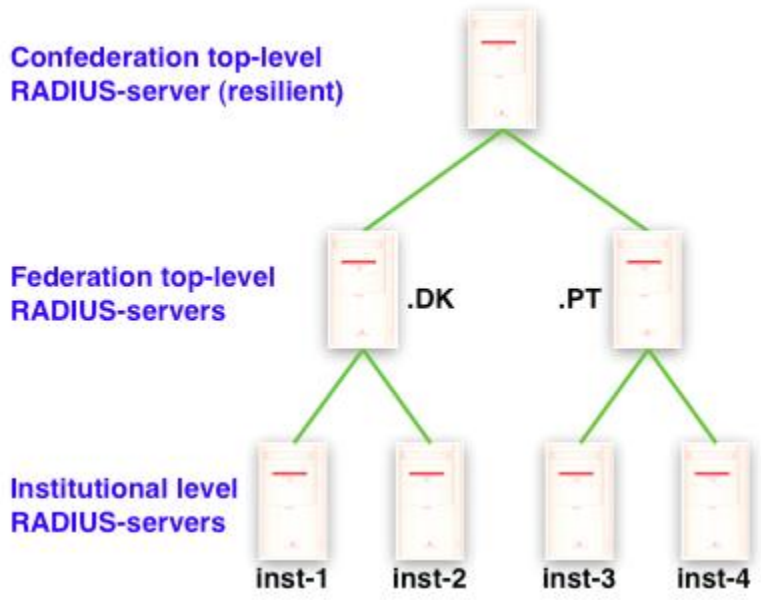


Figura 2.34 – Esquema de xarxa d'Eduroam. Font: www.eduroam.org

Els noms d'usuari acostumen a ser de la forma "user@realm" (on *realm* és el DNS de l'IdP que ha d'autenticar l'usuari). Els servidors RADIUS utilitzen aquest *realm* per encaminar la petició d'autenticació fins a l'IdP adient.

La comunicació entre els *Supplicants* i els punts d'accés i commutadors és realitza mitjançant el protocol 802.1x. Els EAP-Methods acceptats per Eduroam són:

- EAP-PEAP
- EAP-TTLS
- EAP-TLS
- EAP-FAST

Un cop l'usuari s'ha autenticat, l'SP decidirà les polítiques d'autorització a aplicar a l'usuari, que estaran basades en VLANs.

2.3.3 Altres sistemes implantats en organitzacions de la xarxa redIris

Durant la fase de recerca també s'han analitzat altres sistemes de control d'accés a la xarxa implantats per diferents organitzacions de la xarxa redIris:

- *Sistema para el control de acceso a red basado en servicios* - Universidad del País Vasco / Euskal Herriko Unibersitatea (UPV/EHU) [27].
- *Proyecto de Integración de la Tarjeta Inteligente en UPNs* - Universitat de les Illes Balears (UIB) [28].
- *Autenticación centralizada mediante CAS y federación de servicios* - Barcelona Supercomputing Center (BSC) [29].

3. Disseny

En aquest capítol es descriu la fase de disseny del projecte. En primer lloc s'exposen les decisions presses en base a l'estudi realitzat en el capítol anterior. Posteriorment, es detallaran els tipus d'agents (usuaris i dispositius) que participaran en el sistema i la forma en que s'executaran les fases d'autenticació, autorització i *accounting*. Per acabar el capítol, es mostrarà el disseny del pilot, que ha servit per provar totes les funcionalitats del sistema.

3.1 Decisions de disseny

En aquest apartat es justifica l'elecció les tecnologies que seran utilitzades en el sistema de control d'accés a la xarxa. La totalitat de les tecnologies estudiades en el projecte estan descrites en el capítol 2.

Per a desenvolupar el sistema de control d'accés a la xarxa, ha estat necessari prendre les següents cinc decisions de disseny:

- Quin/s protocol/s de control d'accés basat en ports utilitzar?
- Quin protocol AAA utilitzar?
- Quin servidor AAA utilitzar?
- Quin/s mètode/s d'autenticació acceptarà el sistema?
- Quin/s *Supplicant/s* 802.1x utilitzar?

A continuació es dedica un subapartat a respondre cadascuna d'aquestes preguntes. En ells, es descriuran els requisits que ha de complir cada tecnologia candidata i els motius de l'elecció final.

3.1.1 Els protocols de control d'accés basat el port: 802.1x i MAB

Els protocols de control d'accés basat en ports estudiats en la fase de recerca han estat: IEEE 802.1x, VMPS i MAB. La descripció d'aquests protocols es pot trobar en l'apartat 2.1.

A continuació és descriuen els requisits que ha de complir un protocol de control d'accés basat en port per a ser vàlid en el sistema:

- **Autenticació d'usuaris** Com que un dels objectius del sistema de control d'accés a la xarxa és poder autenticar usuaris, és imprescindible disposar d'almenys un protocol que ofereixi aquesta funcionalitat.

- **Autenticació de dispositius** No tots els dispositius que necessiten accedir a la xarxa disposen de sistemes operatius que puguin iniciar i gestionar processos d'autenticació complexos. Les impressores i els telèfons IP en són dos exemples. Per tant, és imprescindible dotar el sistema de control d'accés d'un protocol capaç d'autenticar aquests tipus de dispositius.
- **Compatibilitat amb l'*Authenticator*** Per tal de minimitzar el cost econòmic del projecte, es valorarà que els protocols siguin compatibles amb l'*Authenticator* existent (commutador Cisco Catalyst 3550). D'aquesta forma no caldrà generar despeses relacionades amb l'adquisició de nou hardware.

A continuació s'analitza si els protocols candidats compleixen els requisits:

- **IEEE 802.1x** És un protocol creat per a l'autenticació d'usuaris. És necessari que el dispositiu que vulgui autenticar-se utilitzant aquest protocol disposi d'un *Supplicant* 802.1x. És suportat per l'*Authenticator.*
- **VMPS** Protocol dissenyat per autenticar dispositius utilitzant la seva adreça MAC. No permet l'autenticació d'usuaris. És suportat per l'*Authenticator.*
- **MAB** Protocol dissenyat per autenticar dispositius utilitzant la seva adreça MAC. No permet l'autenticació d'usuaris. És suportat per l'*Authenticator.*

Segons aquest anàlisi i el realitzat en l'apartat 2.1, es poden extreure les següents conclusions:

- El protocol IEEE 802.1x és l'únic que permet l'autenticació d'usuaris, per tant, s'haurà d'utilitzar en el sistema de control d'accés.
- Tant VMPS com MAB únicament autentiquen dispositius.
- Tots tres dispositius són compatibles amb l'*Authenticator* utilitzat. Aquesta conclusió era d'esperar en els casos VMPS i MAB, ja que són protocols desenvolupats per Cisco, fabricant de l'*Authenticator* utilitzat.
- IEEE 802.1x és complementa molt bé amb els protocols AAA gràcies a l'ús dels EAP-Methods.
- VMPS seria un protocol perfectament vàlid per al sistema. Però MAB (evolució de VMPS) proporciona un major nombre de configuracions en l'*Authenticator.*
- A més, MAB està pensat per funcionar conjuntament amb 802.1x; l'autenticació MAB comença quan es detecta que l'autenticació 802.1x no es pot realitzar (figura 2.12).

Per tots aquests motius, **IEEE 802.1x** i **MAB** seran els protocols de control d'accés basat en ports utilitzats en el sistema.

3.1.2 El protocol AAA: RADIUS

Els protocols AAA estudiats en la fase de recerca han estat: RADIUS, TACACS+ i Diameter. La descripció d'aquests protocols es pot trobar en l'apartat 2.2.5.

A continuació és descriuen els requisits que ha de complir un protocol AAA per a ser vàlid en el sistema:

- **Compatibilitat amb el client AAA** Per tal de minimitzar el cost econòmic del projecte, es valorarà que els protocols siguin compatibles amb el client AAA existent (commutador Cisco Catalyst 3550). D'aquesta forma no caldrà generar despeses relacionades amb l'adquisició de nou hardware.
- **Existència de servidors que compleixin els requisits** Un aspecte important a l'hora d'escollir el protocol AAA són els servidors que l'implementen. De res serveix escollir un protocol vàlid per al sistema si cap dels servidors existents per aquest protocol compleix els requisits. Com es veurà més endavant, algunes de les funcionalitats que ha de complir un servidor AAA per a ser vàlid pel sistema són: capacitat per autenticar usuaris en LDAP i Active Directory, suport a gestors de bases de dades, compatibilitat amb els mètodes d'autenticació més segurs i llicència de software lliure.

Gràcies a les taules 2.32 i 2.33 del capítol anterior es pot analitzar si els protocols candidats compleixen els requisits:

- **RADIUS** És un protocol suportat pel client AAA. Existeixen servidors RADIUS capaços d'autenticar usuaris en LDAP i Active Directory, d'emmagatzemar informació varies bases de dades i de suportar l'ús dels mètodes d'autenticació més segurs. Algun d'aquests servidors és de software lliure.
- **TACACS+** És suportat pel client AAA. TACACS+ és un protocol propietari de Cisco. Això obliga a adquirir servidors d'aquest fabricant per a ser utilitzat. A més, el seu servidor (Cisco ACS) no suporta bases de dades de software lliure. Aquests dos aspectes provoquen que el cost econòmic de la seva implantació sigui alt. Per altra banda, Cisco ACS permet l'autenticació d'usuaris simultàniament en Active Directory i LDAP, suportant alguns dels mètodes d'autenticació més segurs.
- **Diameter** No és suportat pel client AAA. Tot i la existència de servidors Diameter de software lliure, actualment cap d'ells és capaç d'autenticar usuaris en cap servei de directori. FreeDiameter permet l'ús de bases de dades per emmagatzemar informació.

Segons aquest anàlisi i el realitzat en l'apartat de la recerca dedicat a AAA, es poden extreure les següents conclusions:

- RADIUS compleix tots els requisits del sistema, i com es veurà en els següents apartats, disposa de servidors de software lliure molt potents.
- TACACS+ també compleix els requisits del sistema. Però a diferència de RADIUS, el cost econòmic de la seva implantació és alt.
- Per tant, RADIUS i TACACS+ són dos protocols vàlids. Però l'ús de RADIUS minimitza el cost del projecte.
- Diameter no compleix la majoria dels requisits del sistema.

Pels motius a dalt exposats, **RADIUS** serà el protocol AAA utilitzat en el sistema.

3.1.3 El servidor RADIUS: freeRADIUS

Un cop escollit RADIUS com a protocol AAA, cal escollir el servidor d'autenticació a utilitza. En les taules 2.32 i 2.33 del capítol anterior es mostren taules representatives dels servidors RADIUS estudiats.

A continuació és descriuen els requisits que ha de complir un servidor RADIUS per a ser vàlid en el sistema:

- **Autenticació en serveis de directori** El servidor ha de ser capaç de poder autenticar usuaris en LDAP i en Active Directory.
- **Suport a EAP-Methods** És imprescindible que el servidor ofereixi suport per algun dels mètodes d'autenticació més segurs (EAP-TLS, EAP-TTLS, EAP-PEAP, EAP-FAST).
- **Software lliure** Per tal de minimitzar el cost econòmic del sistema, es tindrà en consideració el fet que servidor sigui de software lliure.

Analitzant les taules citades i observant els requisits demandats, es poden extreure les següents conclusions:

- Únicament els servidors freeRADIUS, Cisco ACS i Juniper SBR són capaços d'autenticar els usuaris en LDAP i Active Directory.
- Juniper SBR suporta l'ús de tots els mètodes d'autenticació citats com a requisits. freeRADIUS no suporta EAP-FAST. Cisco ACS no suporta EAP-TTLS.
- Dels tres servidors RADIUS, freeRADIUS és l'únic de software lliure. A més, suporta l'ús de bases de dades també de software lliure.

Segons les conclusions a dalt descrites, freeRADIUS és l'únic servidor de software lliure que permet l'autenticació d'usuaris en LDAP i Active Directory, suportant la majoria de mètodes d'autenticació.

Gràcies al gran rendiment que ofereix a cost zero, freeRADIUS és un dels servidors RADIUS més utilitzats del mercat. És un servidor que es troba en un procés de millora constant, i els seus desenvolupadors es mostren molt actius en llistes de distribució, fòrums, wikis, etc.

Per tots aquests motius, s'ha escollit **freeRADIUS** com a servidor RADIUS del sistema. La conseqüència negativa d'aquesta elecció, és que el sistema no suportarà l'ús del mètode d'autenticació EAP-FAST.

En l'Annex I d'aquest document es pot trobar un petit manual de freeRADIUS ideal per a iniciar-se en aquest servidor.

3.1.4 Els mètodes d'autenticació: EAP-PEAP+MSCHAPv2 i EAP-TTLS+MSCHAPv2

Per realitzar la comunicació lògica entre els *Supplicants* 802.1x i freeRADIUS caldrà decidir quins mètodes d'autenticació es voldrà que suporti el sistema. En l'estudi realitzat en l'apartat 2.1.1.3 s'ha conclòs que els EAP-Methods candidats seran aquells que a dia d'avui no ofereixen vulnerabilitats: EAP-TLS, EAP-TTLS, EAP-PEAP i EAP-FAST.

A continuació és descriuen els requisits que ha de complir un mètode d'autenticació per a ser vàlid en el sistema:

- **Compatibilitat amb freeRADIUS** Com que serà el servidor AAA del sistema de control d'accés, és imprescindible que els EAP-Methods acceptats siguin compatibles amb freeRADIUS.
- **Compatibilitat amb varietat de *Supplicants*** Es tindrà en consideració el fet que els mètodes d'autenticació siguin acceptats per diferents *Supplicants* 802.1x. D'aquesta forma, l'elecció del *Supplicant* podrà ser més laxa i no anirà lligada a aquell què suporta l'EAP-Method escollit.

Segons la taula de compatibilitat entre servidors AAA i mètodes d'autenticació (taula 2.33) i la taula de compatibilitat entre *Supplicants* 802.1x i mètodes d'autenticació (taula 2.9), podem treure les següents conclusions:

- freeRADIUS no és compatible amb EAP-FAST. Per tant, aquest mètode d'autenticació no podrà ser acceptat pel sistema.
- Tots els *Supplicants* 802.1x suporten els tres mètodes d'autenticació restants, a excepció de SecureW2, que no suporta EAP-TLS i del *Supplicant* nadiu de Microsoft, que no suporta EAP-TTLS.

Durant l'estudi dels mètodes d'autenticació, s'ha vist que l'EAP-TLS obliga a utilitzar certificats d'usuari. Això significa que en cas d'acceptar-lo com a mètode vàlid, s'haurien de crear (i gestionar) certificats per cadascun dels usuaris als quals se'ls vulgues oferir accés a la xarxa, amb el cost que això suposa. A més, les reconnexions EAP-TLS són lentes i generen un gran trànsit de dades. Degut a aquests motius, s'han decidit descartar EAP-TLS com a mètode d'autenticació en el sistema.

EAP-PEAP i EAP-TTLS funcionen mitjançant la creació d'un túnel TLS per dur a terme les comunicacions. Dins d'aquest túnel cal utilitzar un altre mètode per xifrar les credencials de l'usuari. L'escollit és MSCHAPv2, l'última versió (i més segura) de CHAP.

Per tant, els mètodes d'autenticació que suportarà el sistema d'accés a la xarxa seran **EAP-PEAP+MSCHAPv2** i **EAP-TTLS+MSCHAPv2**.

3.1.5 Els *Supplicants* 802.1x: Microsoft i wpa_supplicant

Com que l'IEEE 802.1x serà un dels protocols d'autenticació basat en ports, caldrà escollir els *Supplicants* 802.1x que s'utilitzaran en les estacions de treball. Es necessitaran *Supplicants* compatibles amb els sistemes operatius Windows XP, Windows 7 i Linux.

Els *Supplicants* 802.1x estudiats en la fase de recerca han estat: el *Supplicant* nadiu de Microsoft, SecureW2, Open1x, wpa_supplicant i Network Manager. El resultats de la recerca es mostren en l'apartat 2.1.1.5.

A continuació és descriuen els requisits que ha de complir un *Supplicants* 802.1x per a ser vàlids en el sistema:

- **Compatibilitat amb els mètodes d'autenticació** El *Supplicant* 802.1x ha de ser compatible amb algun dels EAP-Methods acceptats pel sistema: EAP-PEAP+MSCHAPv2 i EAP-TTLS+MSCHAPv2.

- **Suport a la reutilització de credencials** El *Supplicant* ha de ser capaç de capturar les credencials introduïdes per l'usuari per entrar en el sistema operatiu, i utilitzar-les per autenticar l'usuari a la xarxa de forma transparent.

En la taula 2.9 del capítol anterior es mostra un anàlisi entre els diferents *Supplicants* 802.1x. Observant aquesta taula i els requisits demandats, es poden obtenir les següents conclusions:

- Tots els *Supplicants* 802.1x suporten els dos mètodes d'autenticació requerits, a excepció del *Supplicant* nadiu de Microsoft, que només accepta EAP-PEAP.
- En sistemes Windows XP i Windows 7, únicament el *Supplicant* nadiu de Microsoft és capaç de reutilitzar credencials.
- En sistemes Linux, només l'Wpa_supplicant és capaç de reutilitzar credencials.

Com que accepten algun dels mètodes d'autenticació requerits i són els únic capaços de reutilitzar credencials, s'ha decidit escollir el ***Supplicant* de Microsoft** en sistemes Windows XP i Windows 7, i l'**Wpa_supplicant** en sistemes Linux.

Cal recordar que aquesta decisió només afecta a les estacions de treball. En dispositius externs, el usuaris podran utilitzar el *Supplicant* 802.1x que vulguin, sempre que suportin els EAP-Methods acceptats per el sistema de control d'accés.

3.2 Dispositius

En el sistema hi ha varis tipus de dispositius mitjançant els quals l'usuari pot realitzar peticions d'accés a la xarxa. La font on s'autenticaran els usuaris i els recursos de xarxa que se li oferiran dependran, en part, del dispositiu utilitzat.

Estacions de treball

Les estacions de treball són els ordinadors que la facultat facilita als estudiants i membres de l'LCFIB per dur a terme les seves tasques. Cada estació de treball pertanyerà a un dels quatre perfils següents:

- **Estacions de treball A5** Són els ordinadors que es troben a la planta -1 de l'edifici A5 del campus nord, destinats a donar suport acadèmic als estudiants.
  Utilitzen els sistemes operatius Windows XP i Linux SuSE.
- **Estacions de treball B5** Són els ordinadors que es troben a la planta -1 i -2 de l'edifici B5 del campus nord, destinats a donar suport acadèmic als estudiants.

Utilitzen els sistemes operatius Windows XP i Linux SuSE.

- **Estacions de treball C6** Són els ordinadors que es troben a la planta -3 de l'edifici C6 del campus nord, destinats a donar suport acadèmic als estudiants.
  Utilitzen els sistemes operatius Windows XP i Linux SuSE.
- **Estacions de treball LCFIB** Són els ordinadors que es troben a l'edifici B6 del campus nord, a la planta -1 de l'edifici B5 i en els serveis d'operació dels edificis A5 i C6 (plantes -1 i -3 respectivament). Destinats a ús professional dels membres de l'LCFIB.
  Utilitzen el sistema operatiu Windows 7.

Dispositius externs

Són aquells ordinadors que no són estacions de treball (normalment ordinador portàtils personals). Poden utilitzar qualsevol sistema operatiu.

Dispositius sense *Supplicant*

Són tota la resta de dispositius de xarxa que no són ordinadors (per exemple impressores i telèfons IP). S'han d'autenticar mitjançant MAB perquè no disposen d'un *Supplicant* 802.1x.

***Authenticator* (client RADIUS)**

L'*Authenticator* utilitzat és un commutador CISCO Catalyst 3550. Utilitzarà els protocols 802.1x i MAB en la banda del *Supplicant* i RADIUS en la banda del servidor.

3.3 Usuaris

Cada usuari del sistema té un perfil. Els recursos de xarxa que se li oferiran a cada usuari podrà dependre del perfil:

- **Estudiants** Format pels estudiants de la facultat. Accedeixen a la xarxa mitjançant:
  - Estacions de treball de l'A5, B5 i C6.
  - Dispositius externs.
- **Professors** Format pels professors que imparteixen docència a la facultat. Accedeixen a la xarxa mitjançant:
  - Estacions de treball de l'A5, B5 i C6.

- Dispositius externs.

· **LCFIB** Format pels membres de l'LCFIB. Accedeixen a la xarxa mitjançant:

- Estacions de treball de l'A5, B5 i C6.

- Estacions de treball de l'LCFIB.

- Dispositius externs.

· **Convidats** Es considera un convidat tota persona que no formi part de cap dels perfils anteriors. Accedeixen a la xarxa mitjançant:

- Dispositius externs.

3.4 Autenticació

Segons les circumstancies, els usuaris seran autenticats en un servei de directori LDAP o en un Active Directory. Per a poder ser autenticat, és necessari que l'usuari faciliti el seu nom d'usuari i contrasenya, i que accepti el certificat del servidor.

El servidor RADIUS escollirà el servei de directori on autenticar l'usuari depenent del perfil del dispositiu des d'on l'usuari està realitzant la petició d'autenticació.

L'autenticació en LDAP es realitza sobre aquells usuaris que s'intenten autenticar des de:

- Estacions de treball A5, B5 i C6.

- Dispositius externs.

L'autenticació en Active Directory es realitza sobre aquells usuaris que s'intentin autenticar des de:

- Estacions de treball de l'LCFIB.

3.4.1 Autenticació en LDAP

La FIB té emmagatzemades les dades de tots els seus usuaris en un servei de directori LDAP. Actualment, quan un usuari vol iniciar una sessió en els ordinadors de les estacions de treball A5, B5 i C6, les seves credencials són validades en aquest LDAP.

En el sistema s'utilitzarà el mateix directori LDAP per autenticar els usuaris a la xarxa. El software LDAP utilitzat per realitzar aquesta tasca és openLDAP.

No tots els protocols d'autenticació EAP són compatibles amb tots els tipus de xifrats de contrasenyes. La següent taula mostra la compatibilitat entre els diferents tipus de xifrats de contrasenya i els mètodes d'autenticació suportats per freeRADIUS:

	Clear-text	NT hash (ntlm_auth)	MD5 hash	Salted MD5 hash	SHA1 hash	Salted SHA1 hash	Unix Crypt
PAP	✓	✓	✓	✓	✓	✓	✓
CHAP	✓	x	x	x	x	x	x
Digest	✓	x	x	x	x	x	x
MS-CHAP	✓	✓	x	x	x	x	x
PEAP	✓	✓	x	x	x	x	x
EAP-MSCHAPv2	✓	✓	x	x	x	x	x
Cisco LEAP	✓	✓	x	x	x	x	x
EAP-GTC	✓	✓	✓	✓	✓	✓	✓
EAP-MD5	✓	x	x	x	x	x	x
EAP-SIM	✓	x	x	x	x	x	x

Taula 3.1 – Compatibilitat entre mètodes d'autenticació i xifrats de contrasenyes. Font: www.deployingradius.com

Per saber els formats compatibles amb el mètode d'autenticació escollit cal fixar-se en la fila que fa referència a MS-CHAP, ja que dins dels túnel PEAP i TTLS s'utilitzarà aquest tipus d'autenticació. Com s'observa, els formats de xifrat compatibles són: *Clear-Text* (sense xifrar) i *NT hash (NT Lan Manager Authenticaction)*.

L'LDAP consultat pel Laboratori de Càlcul per autenticar els usuaris de la facultat emmagatzema les contrasenyes en varis formats, entre ells NT hash. Per tant, aquesta restricció no suposarà cap impacte en el disseny del sistema.

3.4.2 Autenticació en Active Directory

Actualment, els membres de l'LCFIB s'autentiquen en un Active Directory per poder accedir a les estacions de treball LCFIB, que formen part d'un domini Windows.

FreeRADIUS utilitzarà aquest mateix Active Directory per autenticar a la xarxa els usuaris connectats des de les estacions de treball LCFIB.

Per a que freeRADIUS autentiqui els usuaris mitjançant Active Directory, és necessari que el servidor també formi part del domini Windows. Per afegir el servidor al domini i poder autenticar a través d'ell cal configurar els serveis Samba i Kerberos.

3.4.3 On autenticar l'usuari

Com s'ha explicat en apartats anteriors, els membres de l'LCFIB poden autenticar-se a la xarxa des de:

- Estacions de treball A5, B5 i C6.
- Dispositius externs.
- Estacions de treball LCFIB.

En els dos primers casos, freeRADIUS autenticarà aquests tipus d'usuaris en l'LDAP. En l'últim cas, ho farà sobre Active Directory.

Per tant, el servidor freeRADIUS ha de ser capaç de deduir (en base al perfil de l'estació de treball que demana accés a la xarxa) en quin servei de directori ha d'autenticar l'usuari.

Per dur a terme aquesta deducció, freeRADIUS s'ajudarà del nom de domini de l'estació de treball; Quan una estació de treball Windows pertany a un domini i utilitza el *Supplicant* nadiu de Microsoft, l'atribut *User-Name* de les trames RADIUS enviades té el format *NomDomini\User-Name.*

Per tant, per saber si l'estació de treball que ha enviat la petició és del perfil LCFIB, freeRADIUS només ha de comprovar l'atribut *User-Name* de les trames RADIUS rebudes. Si el *User-Name* comença pel nom de domini de l'LCFIB, autenticarà l'usuari en Acitve Directory. En la resta de casos, l'autenticació es realitzarà sobre LDAP.

3.5 Autorització

La fase d'autorització es realitzarà en funció del perfil del dispositiu (si és una estació de treball) o del perfil de l'usuari (si és un dispositiu extern).

3.5.1 VLANs

L'autorització es basa en VLANs. Una VLAN és un mètode per crear xarxes lògicament independents dins d'una mateixa xarxa física. Això permet que dos dispositius connectats a un mateix commutador puguin pertànyer a xarxes diferents, i que dos dispositius connectats a diferents commutadors puguin pertànyer a una mateixa xarxa.

En el sistema de control d'accés, a cada VLAN se li oferiran diferents recursos de xarxa. Els *Supplicants* seran assignats a una VLAN concreta en funció del nivell d'autorització que se'ls li vulgui oferir.

Es diferencien dos tipus d'VLANs:

- **D'estació de treball** En aquest grup d'VLANs s'hi col·locaran les estacions de treball. Les polítiques d'autorització es basen en el perfil de l'estació de treball des d'on s'està realitzant la petició.
- **D'usuari** En aquest grup d'VLANs s'hi col·locaran els dispositius externs. Les polítiques d'autorització estaran basades en l'usuari que està realitzant la petició.

3.5.1.1 VLANs d'estacions de treball

En un cas ideal, les estacions de treball A5, B5 i C6 formarien part d'una única VLAN (perquè tindran accés als mateixos recursos de xarxa). Però el gran nombre de dispositius que formen cadascun d'aquests grups provoca que una única subxarxa no tingui suficient adreces IP disponibles per a totes les estacions.

Per aquest motiu, s'ha decidit que cadascun d'aquests grups d'estacions de treball pertanyin a una VLAN diferent.

La següent taula mostra a quina VLAN s'ha associat cada grup d'estacions de treball:

PERFIL	VLAN
Workstations_A5	A5_VLAN
Workstations_B5	B5_VLAN
Workstations_C6	C6_VLAN
Workstations_LCFIB	LCFIB_VLAN

3.5.1.2 VLANs d'usuaris

Aquest tipus d'VLANs estan dedicades a aquells usuaris que es connecten a la xarxa a través de dispositius externs. Es diferencien 5 tipus diferents:

- **STUDENTS_VLAN** Dedicada als estudiants.

- **PROFESSOR_VLAN** Dedicada als professors.
- **PAS_VLAN** Dedicada als membres de l'LCFIB.
- **GUEST_VLAN** Dedicada als usuaris convidats. Quan l'*Authenticator* detecta que algú s'ha connectat a una de les seves interfícies, envia peticions 802.1x preguntant per la identitat de l'usuari. Si l'usuari no respon, se li torna a fer una nova petició.

  Si després d'haver realitzat tres peticions el Supplicant encara no ha respòs, el commutador suposarà que el dispositiu no suporta el protocol 802.1x i internarà autenticar-lo utilitzant MAB. Com que el servidor no té registrada l'adreça MAC del dispositiu extern el procés d'autenticació fallarà i l'usuari serà col·locat en aquesta VLAN, on tindrà accés a uns recursos de xarxa molt reduïts.
- **AUTH_FAIL_VLAN** Dedicada a aquells usuaris que s'autentiquin erròniament. L'usuari té 3 intents consecutius per realitzar l'autenticació. Si no aconsegueix autenticar-se correctament, serà mogut a l'AUTH_FAIL_VLAN. Un cop situat en aquesta VLAN ja no serà capaç de realitzar noves peticions d'autenticació.

La següent taula mostra els diferents grups en que s'han dividit els usuaris, així com l'VLAN associada a cada grup:

PERFIL	VLAN
Estudiants	STUDENTS_VLAN
Professors	PROFESSOR_VLAN
LCFIB	PAS_VLAN
Convidats	GUEST_VLAN
Aut. fallida	AUTH_FAIL_VLAN

3.5.1.3 Diferenciar entre peticions des d'estacions de treball i des d'ordinadors externs

Per aplicar el tipus d'autorització adient, freeRADIUS necessita identificar si les peticions d'autorització rebudes provenen d'una estació de treball o d'un dispositiu extern.

Per fer-ho, s'han emmagatzemat en una base de dades les adreces MAC de totes les estacions de treball. També s'emmagatzema una relació entre l'estació de treball i el seu perfil.

Quan el servidor RADIUS rep una petició d'autorització, comprova l'adreça MAC del Supplicant (atribut *Calling-Station-Id* de la trama RADIUS). Si el servidor troba l'adreça MAC a la base de dades, suposarà que la petició s'envia des d'una estació de treball i li aplicarà les polítiques d'autorització en funció del seu perfil.

Si l'adreça MAC no apareix a la base de dades, el servidor RADIUS deduirà que l'usuari no s'està autenticant des de cap estació de treball i li aplicarà l'autorització basant-se en el perfil de l'usuari.

Tot i que en el procés d'autorització el servidor s'ajuda de l'adreça MAC del dispositiu, cal notar que aquest tipus d'autorització no es MAB, ja que el Supplicant es comunica amb l'*Authenticator* utilitzant trames 802.1x.

3.5.1.4 VLANs d'arrencada

Totes les estacions de treball s'ajuden de la xarxa per arrencar. Per poder tenir accés a la xarxa, cal realitzar l'autenticació 802.1x. Però l'autenticació 802.1x no es pot realitzar fins que el sistema operatiu no s'ha iniciat. I el sistema operatiu no es pot iniciar si el dispositiu no pot arrencar. És un peix que es mossega la cua.

La solució adoptada en aquest cas ha estat utilitzar l'autenticació MAB enlloc de l'autenticació 802.1x, ja que l'autenticació MAB es pot dur a terme abans que arranqui el sistema operatiu.

Interessa que aquest tipus d'autenticació es realitzi el més aviat possible. Tant bon punt arrenca el dispositiu, l'*Authenticator* haurà de capturar l'adreça MAC del dispositiu i realitzar el procés d'autenticació MAB.

Dos paràmetres en el commutador Cisco permeten definir el temps que ha de trigar l'*Authenticator* en descartar l'autenticació 802.1x i començar l'autenticació MAB:

- **max-reauth-req** Indica el màxim nombre de cops que l'*Authenticator* intentarà realitzar l'autenticació 802.1x enviant un paquet *EAP-Request-Identity*.
- **tx-period** Indica el temps d'espera (en segons) entre l'enviament de dos *EAP-Request-Identity*.

La fórmula que utilitza el commutador per calcular el període de temps que ha de passar per descartar l'autorització 802.1x i començar l'autorització MAB és la següent:

[(max-reauth-req + 1) * tx-period]

Assignant els paràmetres *max-reauth-req* i *tx-period* a 1 (mínim valor possible) es pot aconseguir que l'*Authenticator* descarti l'autenticació 802.1x i comenci l'autenticació MAB al cap de 2 segons. Configurar un temps inferior és impossible.

Utilitzant MAB, el servidor RADIUS pot saber si el dispositiu és una estació de treball, i com a tal, necessita recursos de xarxa per arrencar. Per fer-ho, comprovarà en una base de dades si la MAC del dispositiu correspon a la d'una estació de treball.

Igual que passava en el cas de les autoritzacions d'estacions de treball, s'han hagut de crear varies VLANs d'arrencada per donar suport al gran nombre d'estacions de treball.

La següent taula mostra els perfils en que s'han dividit les diferents estacions de treball, així com l'VLAN associada a cada un d'ells:

PERFIL	VLAN
workstations_A5	A5_BOOT_VLAN
workstations_B5	B5_BOOT _VLAN
workstations_C6	C6_BOOT _VLAN
workstations_LCFIB	LCFIB_BOOT _VLAN

3.5.2 El procés d'autorització

El següent diagrama de flux resumeix el procediment que es dur a terme per realitzar l'autorització a la xarxa:

Figura 3.2 – Procés d'autorització del sistema de control d'accés a la xarxa

3.5.3 Emmagatzemament de les polítiques d'autorització

En aquest apartat es mostra la forma en que es guarden les dades relacionades amb el procés d'autorització.

3.5.3.1 Atributs RADIUS d'autorització

Per tal d'assignar les polítiques d'autorització, freeRADIUS enviarà els següents atributs RADIUS a l'*Authenticator*, indicant-li en quina VLAN s'ha de col·locar cada dispositiu:

Tunnel-Medium-Type = IEEE-802
Tunnel-Type = VLAN
Tunnel-Private-Group-ID = "VLAN_NAME" or "VLAN NUMBER"

3.5.3.2 La base de dades

Per emmagatzemar tota la informació d'autenticació i autorització mitjançant adreces MAC s'utilitzarà una base de dades MySQL. FreeRADIUS facilita un SCHEMA SQL del qual s'utilitzen les següents taules:

- **Usergroup** Permet associar usuaris a grups (perfils).

Field	Type	Null	Key	Default	Extra
Username	varchar (64)	NO	MUL		
groupname	varchar (64)	NO			
priority	int (11)	NO		1	

- **Radgroupreply** Permet configurar l'autorització indicant quins atributs RADIUS s'enviaran en les trames de resposta a l'*Authenticator*. Notar que aquesta taula ens permet configuració per grups (perfils), no per usuaris ni dispositius.

Field	Type	Null	Key	Default	Extra
Id	int (11) unsigned	NO	PRI	NULL	auto_increment
groupname	varchar (64)	NO	MUL		
attribute	varchar (64)	NO			
op	char (2)	NO		=	
value	varchar (253)	NO			

En el nou sistema es necessitaran dos taules *radgroupreply*: una on s'emmagatzemaran les polítiques d'autorització d'arrencada, i un altra per emmagatzemar les polítiques d'autorització globals de les estacions de treball.

Per tant, caldrà modificar l'SCHEMA SQL per afegir una nova taula *radgroupreply* que s'anomenarà *authorized_boot_VLAN*. Per temes d'estandardització de noms també caldrà canviar el nom de la taula *radgroupreply* original pel de *authorized_workstation_VLAN*.

D'aquesta forma s'utilitzaran tres taules SQL per gestionar l'autenticació i autorització d'estacions de treball:

- **Usergroup** Associa una estació de treball a un perfil.
- **Authorized_boot_VLAN** Autoritza una estació de treball a arrencar per xarxa.
- **Authorized_workstation_VLAN** Autoritza de forma general una estació de treball.

Es crearan dos mòduls freeRADIUS per dur a terme els processos d'autorització basats en bases de dades:

- **is_authorized_to_boot_from_network** Autoritza les estacions de treball a arrencar per xarxa. Primer comprova si el dispositiu és a la taula *usergroup*. En cas afirmatiu, mira el seu perfil i aplica les polítiques d'autorització d'aquest perfil definides a la taula *authorized_boot_vlan*.
- **is_workstation** Autoritza les estacions de treball per accedir a la xarxa. Primer, comprova si el dispositiu és a la taula *user_group*. En cas afirmatiu, mira el seu perfil i aplica les polítiques d'autorització d'aquest perfil definides a la taula *authorized_workstation_vlan*.

3.5.3.3 El fitxer users

En el fitxer *users* de freeRADIUS s'emmagatzemaran les polítiques d'autorització basades en els usuaris. En l'Annex I es pot trobar un manual dedicat a freeRADIUS.

3.6 Accounting

El servidor freeRADIUS també gestionarà l'accounting. Per fer-ho, s'ajudarà de la mateixa base de dades utilitzada per gestionar les polítiques d'autorització.

En el mateix SCHEMA SQL de freeRADIUS es troba la següent taula dedicada a l'accounting:

Field	Type	Null	Key	Default	Extra
RadAcctId	bigint(21)	NO	PRI		auto_increment
AcctSessionId	varchar(32)	NO	MUL		
AcctUniqueId	varchar(32)	NO	MUL		
UserName	varchar(64)	NO	MUL		
Realm	varchar(64)	YES			
NASIPAddress	varchar(15)	NO	MUL		
NASPortId	varchar(15)	YES		NULL	
NASPortType	varchar(32)	YES	MUL	NULL	
AcctStartTime	Datetime	NO	MUL		
AcctStopTime	Datetime	NO			
AcctSessionTime	int(12)	YES		NULL	
AcctAuthentic	varchar(32)	YES		NULL	
ConnectInfo_start	varchar(50)	YES		NULL	
ConnectInfo_stop	varchar(50)	YES		NULL	
AcctInputOctets	bigint(12)	YES		NULL	
AcctOutputOctets	bigint(12)	YES		NULL	
CalledStationId	varchar(50)	NO			
CallingStationId	varchar(50)	NO			
AcctTerminateCause	varchar(32)	NO	MUL		
ServiceType	varchar(32)	YES		NULL	
FramedProtocol	varchar(32)	YES		NULL	
FramedIPAddress	varchar(15)	NO			
AcctStartDelay	int(12)	YES		NULL	
AcctStopDelay	int(12)	YES		NULL	

S'ha creat un nou mòdul RADIUS anomenat *sql_accounting* que durà a terme el procés d'emmagatzemament de les dades d'*accounting* a la base de dades.

3.7 Esquema de la xarxa

En la següent figura es mostra l'esquema de xarxa del sistema:

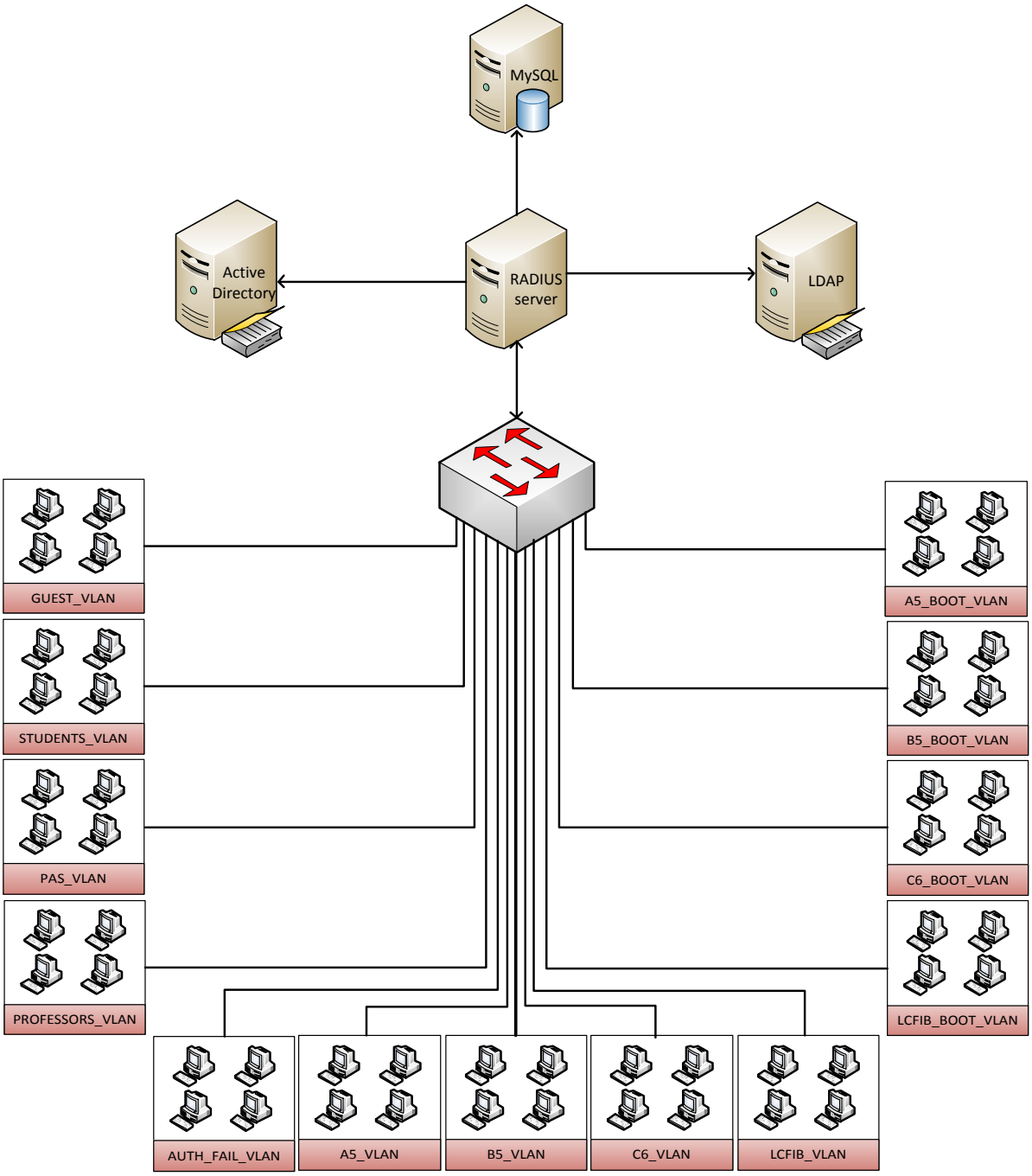


Figura 3.3 – Esquema de xarxa del sistema de control d'accés

3.8 Disseny del pilot

Abans de dur a terme la implementació real del sistema s'ha creat una prova pilot. La prova pilot és una simulació a escala més reduïda del que serà finalment el sistema. L'objectiu del pilot es verificar totes les funcionalitats del sistema i detectar possibles problemes i errors.

En aquest apartat s'explica el disseny del pilot. Totes les decisions de disseny justificades anteriorment en aquest capítol han estat respectades. També es respecta la forma en que s'executen les fases d'autenticació, autorització i accounting. L'únic que varia el nombre de components que formen la xarxa.

3.8.1 Dispositius del pilot

En aquest apartat es descriuen els dispositius utilitzats en la prova pilot:

- ***Supplicant*** És un ordinador de sobretaula on s'hi han instal·lat els sistemes operatius Linux OpenSuSE, Microsoft Windows XP i Microsoft 7. El *Supplicant* ha estat configurat de varies formes per a poder verificar el funcionament de tots els casos possibles d'autenticació i autorització descrits en aquest capítol. Per tant, durant la fase de proves el *Supplicant* tindrà el perfil de:

  - Estació de treball LCFIB.

  - Estació de treball A5.

  - Dispositiu extern.

- ***Authenticator*** **(o client RADIUS)** És un commutador Cisco Catalyst 3550 que s'encarrega de traslladar les peticions de l'usuari a l'*Authentication Server*, d'aplicar les polítiques d'autorització i de generar dades estadístiques sobre l'ús que ha fet l'usuari dels recursos oferts. És qui permet l'accés del *Supplicant* a la xarxa.
- ***Authentication Server*** És un ordinador de sobretaula on s'hi ha instal·lat el sistema operatiu Linux OpenSuSE. S'encarrega d'autenticar els usuaris, decidir quins recursos de xarxa se li ofereixen i emmagatzemar les dades estadístiques sobre l'ús que es fan del recursos. En l'*Authentication Server* si ha instal·lat una base de dades MySQL i un servei de directori LDAP openLDAP.
- **Autenticador Windows** És un ordinador de sobretaula on s'hi ha instal·lat un sistema operatiu Microsoft Server 2008 i un Active Directory. S'encarrega d'autenticar els

usuaris membres de l'LCFIB. S'ha creat el domini Windows *AD,* del qual formarà part el *Supplicant* quan jugui el rol d'estació de treball LCFIB.

La següent figura mostra la distribució dels dispositius que formen el pilot:

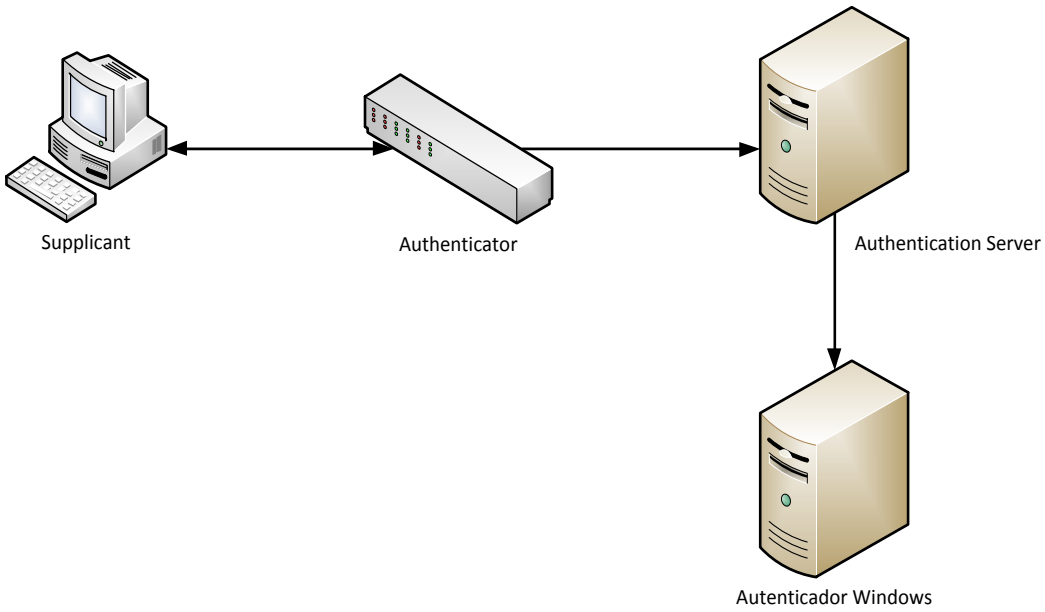


Figura 3.4 – Esquema de xarxa del pilot

3.8.2 Usuaris del pilot

En cada sistema operatiu del *Supplicant* s'han creat tres usuaris:

- **oriol.bellet** És un usuari amb perfil *Estudiants*.
- **prof** És un usuari amb perfil *Professors* .
- **asac** És un usuari amb perfil *LCFIB*.

3.8.3 Autenticació en el pilot

El procés d'autenticació en el pilot es realitzarà de la mateixa manera que la descrita al llarg del capítol. El usuaris seran autenticats en un LDAP o en un Active Directory depenent del perfil del dispositiu des del qual s'està realitzant la petició.

Per a poder realitzar les proves d'autenticació sobre LDAP s'ha creat la següent estructura LDAP, on s'han afegit els usuaris *oriol.bellet*, *prof i asac* descrits anteriorment.

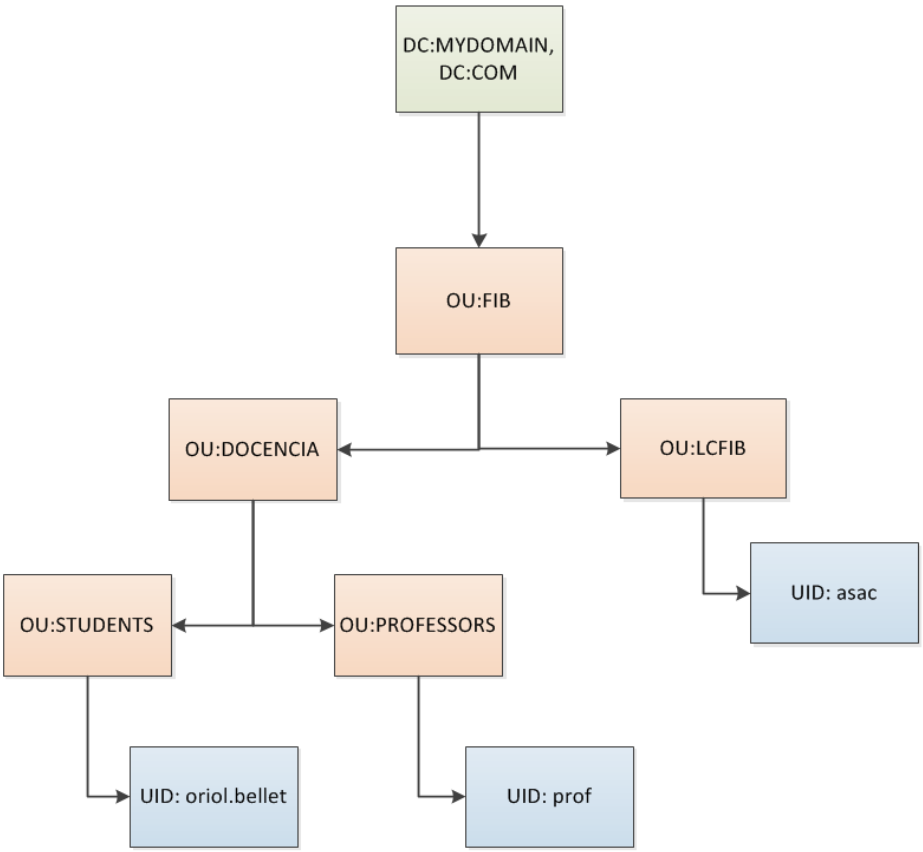


Figura 3.5 – Jerarquia LDAP utilitzada en el pilot

3.8.4 Autorització en el pilot

La fase d'autorització en el pilot també es realitza mitjançant VLANs. S'ha fet una selecció de tots els tipus d'autorització descrits en aquest capítol. Aquesta selecció és suficientment significativa com per permetre provar tots els tipus d'autorització possibles. Els perfils d'autorització escollits són:

- **A5_VLAN** Autoritzarà al *Supplicant* quan aquest estigui configurat com a estació de treball A5. Aquesta cas engloba les VLANs B5_VLAN, C6_VLAN i LCFIB_VLAN, ja que el procés d'autorització és el mateix en totes elles.
- **A5_BOOT_VLAN** Autoritzarà el *Supplicant* a arrencar per xarxa quan aquest estigui configurat com a estació de treball A5. Aquesta cas engloba les VLANs B5_BOOT_VLAN, C6_BOOT_VLAN i LCFIB_BOOT_VLAN, ja que el procés d'autorització és el mateix en totes elles.

- **STUDENTS_VLAN** Autoritzarà al *Supplicant* quan aquest estigui configurat com a dispositiu extern i l'usuari que inicïi sessió sigui *oriol.bellet*.
- **PAS_VLAN** Autoritzarà al *Supplicant* quan aquest estigui configurat com a dispositiu extern i l'usuari que inicïi sessió sigui *asac*.
- **PROFESSORS_VLAN** Autoritzarà al *Supplicant* quan aquest estigui configurat com a dispositiu extern i l'usuari que inicïi sessió sigui *prof*.
- **GUEST_VLAN** Autoritzarà al *Supplicant* quan aquest estigui configurat com a dispositiu extern i l'usuari no realitzi l'autenticació 802.1x.
- **AUTH_FAIL_VLAN** Negarà l'accés del *Supplicant* a la xarxa quan aquest intenti autenticar-se de forma errònia tres cops seguits.

3.8.5 Accounting en el pilot

La fase d'accounting en el pilot serà exactament la mateixa que en el sistema real.

3.9 Vulnerabilitats del sistema

Tot i que el sistema de control d'accés permet augmentar el nivell de seguretat de la xarxa, no està lliure de vulnerabilitats. A continuació es descriuen possibles errors que podrien succeir i la forma de prevenir-los:

- **Els punts d'accés depenen d'un servei** El correcte funcionament del sistema de control d'accés està inevitablement condicionat al correcte funcionament del servidor RADIUS. Una caiguda del servidor provocaria que els usuaris no es poguessin autenticar, i per tant, no podrien accedir a la xarxa. La millor forma d'evitar aquesta situació és disposar d'una rèplica del servidor RADIUS. Els commutadors són capaços de tenir definits varis servidors. Així, quan un commutador detectés que el servidor primari ha caigut, podria enviar les peticions a un servidor secundari, fent que la xarxa no es veies afectada.
- **Un servei de xarxa depèn d'altres serveis de xarxa** El correcte funcionament del servidor RADIUS també depèn d'altres components de la xarxa, com poden ser els serveis de directori o les bases de dades. La vulnerabilitat és similar a la descrita en el punt anterior. Un caiguda d'algun d'aquests serveis faria que el servidor RADIUS no pogués realitzar correctament alguna de les etapes AAA. Per solucionar aquest problema també s'utilitza la tècnica de replicació; tan les bases de dades com els serveis de directori estan duplicats. D'aquesta forma, si el servei principal cau,

freeRADIUS podrà comunicar-se amb el servidor secundari sense que el rendiment de la xarxa es vegi afectat.

- **Suplantació de l'adreça MAC d'una estació de treball** Un usuari malintencionat podria esbrinar l'adreça MAC de la targeta de xarxa d'una estació de treball. Un cop obtinguda, podria canviar l'adreça MAC del seu ordinador portàtil per la MAC capturada i connectar el dispositiu a la xarxa. La conseqüència d'això seria que quan l'usuari inicies el seu ordinador l'*Authenticator* realitzaria l'autenticació MAB per saber si el dispositiu és una estació de treball i, com a tal, necessita accés a la xarxa per arrencar. A l'haver suplantat l'adreça MAC d'una estació de treball, el servidor col·locaria el dispositiu en una de les VLANs dedicades a l'arrencada d'estacions de treball. Per tant, l'usuari obtindria accés a recursos els quals no està autoritzat a utilitzar. Per aquest tipus de vulnerabilitat (ja existent en el sistema actual) no hi ha cap solució possible. Es podria pensar en l'ús de certificats en les estacions de treball, però l'autenticació MAB es realitza abans d'iniciar el sistema operatiu, i el gestors d'arrancada no suporten un intercanvi de certificats a aquest nivell.
- **Usurpació de les credencials d'un usuari** Un usuari malintencionat podria esbrinar el nom d'usuari i contrasenya d'un altre usuari i suplantar la seva identitat en el procés d'autenticació. Aquest fet també passaria desapercebut pel servidor. Una de les solucions radica en l'ús de certificats d'usuari en el procés d'autenticació (funcionalitat opcional en PEAP i EAP-TTLS). Però el cost d'aquesta solució és elevat, ja que s'haurien de crear (i gestionar) certificats per a tots els usuaris. En aquest cas, la millor solució és conscienciar els usuaris sobre la importància de mantenir el secretisme de les seves credencials i seguir les recomanacions de bon ús. Alguns dels consells àmpliament coneguts per tothom són:

  Utilitzar contrasenyes segures (amb dígits, caràcters i lletres).

  No utilitzar contrasenyes fàcilment predictibles.

  No utilitzar la mateixa contrasenya per accedir a varis serveis.

  Canviar la contrasenya periòdicament.
- **Atacs de diccionari** Un usuari malintencionat podria utilitzar atacs de diccionari o de força bruta per intentar esbrinar la contrasenya d'un altre usuari. El protocol 802.1x permet prevenir aquesta situació, ja que quan un usuari falla X cops consecutius en el procés d'autenticació, pot ser col·locat en una de les anomenades *auth_fail_VLAN.* El commutador no permet accedir a la xarxa ni realitzar noves peticions d'autenticació als usuaris que pertanyen a aquesta VLAN

4. Implementació

Aquest capítol està dedicat a descriure com s'ha realitzat la implementació del pilot. Seguint els passos aquí descrits s'hauria de poder realitzar la implementació real del sistema sense cap problema, ja que totes les funcionalitats del sistema han estat testejades en el pilot.

Al llarg del capítol s'aniran mostrant varis quadres explicatius sobre comandes a utilitzar i fitxers a modificar. Els quadres amb fons negre i lletres verdes fan referencia a comandes a executar en un terminal Linux, ja sigui en el *Supplicant*, en el commutador o en el servidor. Els quadres amb fons blau fan referencia als fitxers de configuració a modificar. Les línies a editar en cada fitxer estan remarcades en **negreta**.

4.1 Instal·lació de freeRADIUS

A continuació s'expliquen els requisits necessaris i els passos a realitzar per instal·lar freeRADIUS en l'*Authentication Server*.

4.1.1 Prerequisits

Abans d'instal·lar freeRADIUS cal comprovar que els següents paquets es troben instal·lats en el sistema. En cas de no ser així, es poden obtenir mitjançant els gestor de paquets de la distribució Linux corresponent.

- *make (GNU-Make)*
- *openldap2-client (The OpenLDAP commandline client tools)*
- *openldap2-devel (Libraries, Header Files and Documentation for OpenLDAP)*
- *mysql (A True Multiuser, Multithreaded SQL Database Server)*
- *mysql-client (MySQL Client)*
- *libmysqlclient-devel (MySQL Development Header Files and Libraries)*
- *libopenssl-devel (Include Files and Libraries mandatory for Development)*
- *libopenssl0_9_8 (Secure Sockets and Transport Layer Security)*
- *openssl (Secure Sockets and Transport Layer Security)*

4.1.2 Instal·lació

Creem el directori */home/soft/freeradius* on s'instal·larà el servidor i li donem a l'usuari *radius* la seva propietat. Ja com a usuari *radius*, descarreguem i instal·lem freeRADIUS:

```
$>su
$>mkdir -p /home/soft/freeradius-2.1.10/src
$>ln –s /home/soft/freeradius-2.1.10 /home/soft/freeradius
$>chown –R  radius:radius /home/soft/freeradius-2.1.10
$>su –radius
$>cd /home/soft/freeradius-2.1.10/src
$>wget ftp://ftp.freeradius.org/pub/freeradius/freeradius-server-2.1.10.tar.gz
$>tar xvzf freeradius-server-2.1.10.tar.gz
$>cd freeradius-server-2.1.10
$>./configure --prefix=/home/soft/freeradius --with-openssl --with-openssl-include=/usr/include/openssl --with-openssl-lib=/usr/lib --with-mysql-lib-dir=/home/soft/mysql/lib --with-mysql-include-dir=/home/soft/mysql/include
$>make
$>make install
```

Per posar en marxa el *daemon* de freeRADIUS (com usuari *radius*):

```
$>/home/soft/freeradius/sbin/radiusd
```

Si es vol iniciar en mode *debug* (recomanat per realitzar les proves) cal afegir l'opció *-X*

4.2 Configuració dels clients RADIUS

Tot seguit es mostra com configurar els clients RADIUS (*Authenticators*) per a que puguin interactuar tant amb el servidor freeRADIUS com amb el *Supplicant*.

La configuració es divideix en dues parts. La primera es realitza en el mateix client i la segona en el servidor freeRADIUS.

4.2.1 Configuració del client RADIUS

Per configurar el commutador, ens connectem a ell des d'un terminal:

```
$> telnet 192.168.100.100
Trying 192.168.100.100...
Connected to 192.168.100.100.
Escape character is '^]'.

User Access Verification

Username: admin
Password:

Commutador >enable
Password:
Commutador #conf t
```

Creem totes les VLANs que formen la xarxa. En el següent exemple es mostra la creació de l'A5_ VLAN, on li assignem un nom, un número i una adreça IP:

```
Commutador(config)# interface VLAN 50
Commutador(config-if)# ip address 192.168.50.100 255.255.255.0
Commutador(config-if)# exit
Commutador(config)# vlan 50
Commutador(config-vlan)# name A5_VLAN
```

Seguirem el mateix procediment per crear la resta d'VLANs.

A continuació, configurem el commutador per a que utilitzi AAA. Li definim les polítiques de autenticació, autorització i *accounting*:

```
Commutador (config)#aaa new-model
Commutador (config)#aaa authentication dot1x default group radius enable
Commutador (config)#aaa authorization network default group radius
Commutador (config)#aaa accounting dot1x default start-stop group radius
```

També cal indicar-li al commutador la IP del servidor RADIUS, els ports d'autenticació/autorització i *accounting* i el *shared secret* entre ell i el servidor:

```
Commutador (config)#radius-server host 192.168.100.200 auth-port 1812 acct-port 1813 key
$CommutadorFreeradiusSharedSecret
```

Les interfícies sobre les quals es vol aplicar 802.1x o MAB s'han de configurar una a una. A continuació es mostra la configuració de la interfície *ethernet* 1. Seleccionem la interfície:

```
Commutador(config)# interface f0/1
```

Habilitem 802.1x:

```
Commutador(config-if)# dot1x port-control auto
```

Habilitem MAB:

```
Commutador(config-if)# dot1x mac-auth-bypass
```

Configurem els paràmetres *tx-period* i *max-reauth-req* per a que el commutador comenci l'autenticació MAB quan el *Supplicant* arrenqui:

```
Commutador(config-if)#dot1x timeout tx-period 1
Commutador(config-if)#dot1x max-reauth-req 1
```

Indiquem quina és la Guest-VLAN (caldrà haver-la definit anteriorment) per a aquesta interfície:

```
Commutador(config-if)#dot1x guest-vlan GUEST_VLAN
```

Per últim, li indiquem quina és la *auth-fail* VLAN i el nombre d'intents d'autenticació que té l'usuari abans de ser mogut a ella.

```
Commutador(config-if)#dot1x auth-fail max-attempts 3

Commutador(config-if)#dot1x auth-fail vlan AUTH_FAIL_VLAN
```

4.2.2 Configuració del client en freeRADIUS

La configuració dels clients RADIUS es fa en el fitxer *etc/raddb/clients*. Simplement cal afegir la IP del client, el *shared secret* i el fabricant.

```
client 192.168.100.100 {
        secret = $CommutadorFreeradiusSharedSecret
        nastype = cisco
}
```

4.3 Configuració EAP en freeRADIUS

Els EAP methods sobre els quals s'encapsularan les trames RADIUS es configuren en el fitxer *etc/raddb/eap.conf*. Cal recordar que els mètodes d'autenticació escollits són EAP-PEAP+MSCHAPv2 i EAP-TTLS-MSCHAPv2.

Haurem de comentar les crides dedicades als mètodes d'autenticació que no voldrem que el servidor suporti (MD5, GTC i LEAP). És important no eliminar les entrades dedicades a TLS ja que els mòduls PEAP i TTLS l'utilitzen per configurar el túnel TLS. Tot i que deixant l'EAP-Method habilitat pot semblar que els *Supplicants* podran utilitzar-lo com a mètode d'autenticació, realment no serà així, ja que per utilitzar EAP-TLS els *Supplicants* necessitaran un certificat de client que no tindran.

També modificarem el paràmetre *default_eap_type* i habilitarem el paràmetre *use_tunneled_repy* en PEAP i TTLS. *use_tunneled_repy* permet que el servidor pugui enviar atributs RADIUS al client quan la comunicació es troba dins d'un túnel TLS.

```
[…]
```

```
eap {
        [...]
        default_eap_type = peap
        [...]
        #
        #  We do NOT recommend using EAP-MD5 authentication
        #  for wireless connections.  It is insecure, and does
        #  not provide for dynamic WEP keys.
        #
        #md5 {
        #}
        [...]
        # As a result, LEAP *requires* access to the plain-text User-Password, or the NT-
        # Password attributes. 'System' authentication is impossible with LEAP.
        #leap {
        #}
        [...]
        #  Generic Token Card.
        #  Currently, this is only permitted inside of EAP-TTLS, or EAP-PEAP.  The module
        #"challenges" the user with text, and the response from the user is taken to be the
        #User-Password.
        #Proxying the tunneled EAP-GTC session is a bad idea, the users password will go over
        #the wire in plain-text, for anyone to see.
        #gtc {
                [...]
        #}
        [...]
        ttls {
                [...]
                default_eap_type = mschapv2
                [...]
```

```
                use_tunneled_reply = yes
                [...]
        }
        peap{
                [...]
                use_tunneled_reply = yes
                [...]
        }
}
```

En el pilot s'utilitzaran els certificats d'exemple generats per freeRADIUS, que es troben en el directori *etc/raddb/certs*.

4.4 Autenticació

En els següents apartats s'explica com configurar els sistema per poder autenticar els usuaris tant en LDAP com en Active Directory.

4.4.1 LDAP

Per autenticar els usuaris en LDAP cal configurar *openLDAP* i el mòdul LDAP de freeRADIUS.

4.4.1.1 openLDAP

El primer que cal fer un cop instal·lat *openLDAP* és crear la contrasenya de *root* de l'LDAP. Per major seguretat, la contrasenya s'emmagatzemarà amb xifrat SHA en el fitxer de configuració *d'openLDAP*. Per xifrar la contrasenya s'utilitzarà *slappaswd*, una eina que ve incorporada amb *openLDAP*:

```
$>slappasswd -h {sha}
New password:
Re-enter new password:
{SHA}5en6G6MezRroT3XKqkdPOmY/BfQ=
```

La contrasenya xifrada en SHA és: 5en6G6MezRroT3XKqkdPOmY/BfQ=

Editem el fitxer de configuració global *d'openLdap /etc/openldap/slapd.conf*. Definim el *Domain Component* i la contrasenya LDAP abans creada:

```
#######################################################################
# BDB database definitions
#######################################################################

database        bdb
#The DN suffix of queries that will be passed to this backend database.
suffix          "dc=mydomain,dc=com"
checkpoint      1024    5
cachesize       10000
#The DN that is not subject to access control or administrative limits restrictions for operations on this database.
rootdn          "cn=admin,dc=mydomain,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          {SHA}5en6G6MezRroT3XKqkdPOmY/BfQ=
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory       /var/lib/ldap
# Indices to maintain
index   objectClass     eq
```

Com que en l'LDAP s'utilitzaran atributs RADIUS (*radiusGoupName*), cal afegir l'SCHEMA LDAP facilitat per freeRADIUS a la llista *d'Schemes d'openLDAP*:

```
$>cp /home/soft/freeradius/share/doc/reeradius/examples/openldap.schema
/etc/openldap/schema/RADIUS-LDAPv3.schema
```

Incloem el nou SCHEMA al fitxer de configuració *d'openLDAP /etc/openldap/slapd.conf*.

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/rfc2307bis.schema
include         /etc/openldap/schema/yast.schema
include         /etc/openldap/schema/RADIUS-LDAPv3.schema
[…]
```

En l'LDAP també s'utilitzaran atributs Samba (*sambaNTpassword* i *sambaLMpassword*). Per tant, cal incloure l'SCHEMA de Samba per a LDAP. Aquest SCHEMA es troba en el codi font de SAMBA i es pot obtenir de la següent manera:

```
$> cd /tmp
$>wget http://www.samba.org/samba/ftp/stable/samba-3.5.9.tar.gz
$>tar xvzf samba-3.5.9.tar.gz
$>cp samba-3.5.9/examples/LDAP/samba.schema /etc/openldap/schema/samba3.schema
```

Cal incloure el nou SCHEMA en el fitxer de configuració *d'openLDAP /etc/openldap/slapd.conf*:

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
```

```
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/rfc2307bis.schema
include         /etc/openldap/schema/yast.schema
include         /etc/openldap/schema/RADIUS-LDAPv3.schema
include         /etc/openldap/schema/samba3.schema
[…]
```

Per generar contrasenyes en format NT hash es pot utilitzar l'eina *mkntpwd*. Descarreguem el codi, el compilem i executem el binari generat:

```
$>wget http://www.nomis52.net/data/mkntpwd.tar.gz
$>tar xvzf mkntpwd.tar.gz
$>cd mkntpwd
$>make
$> ./mkntpwd -L oriolBelletLdapPsswd -N oriolBelletLdapPsswd
41557DF84AF0532239C0CAA4C7EE62EB:9DF9D1C13D0A9BCDE5BFB4DF96828074
```

En aquest exemple s'ha xifrat la cadena *oriolBelletLdapPsswd* en format NT hash. El programa retorna per la sortida estàndard les contrasenyes en format LM i NT respectivament separades pel símbol *:* (dos punts).

El següent pas es crear l'estructura de directoris LDAP. Per fer-ho es genera el següent fitxer anomenat *fib.ldif*:

```
# Domain Component

dn: dc=mydomain,dc=com

objectClass: dcObject

objectClass: organizationalUnit

ou: Mydomain.com Radius

dc: mydomain


# Organizational unit fib

dn: ou=fib,dc=mydomain,dc=com

objectclass: organizationalunit

ou: fib


# Organizational unit docencia

dn: ou=docencia,ou=fib,dc=mydomain,dc=com

objectclass: organizationalunit

ou: docencia


# Organizational unit students

dn: ou=students,ou=docencia,ou=fib,dc=mydomain,dc=com

objectclass: organizationalunit

ou: students


# Organizational unit professors

dn: ou=professors,ou=docencia,ou=fib,dc=mydomain,dc=com

objectclass: organizationalunit

ou: professors


# Organizational unit lcfib

dn: ou=lcfib,ou=fib,dc=mydomain,dc=com

objectclass: organizationalunit
```

```
ou: lcfib

# User oriol.bellet (Group gStudents)
dn: uid=oriol.bellet,ou=asac,ou=lcfib,ou=fib,dc=mydomain,dc=com
objectclass: radiusProfile
objectclass: radiusObjectProfile
objectclass: sambaSamAccount
uid: oriol.bellet
cn: oriol.bellet
sambaLmPassword:0x41557DF84AF0532239C0CAA4C7EE62EB
sambaNtPassword:0x9DF9D1C13D0A9BCDE5BFB4DF96828074
radiusGroupName: gStudents
sambaSID: S-1-0-0-28976

# User prof (Group gProfessors)
dn: uid=prof,ou=docencia,ou=fib,dc=mydomain,dc=com
objectclass: radiusProfile
objectclass: radiusObjectProfile
objectclass: sambaSamAccount
uid: prof
cn: prof
sambaLmPassword:0x4A76F587DC294D1F93E28745B8BF4BA6
sambaNtPassword:0x6B6BEE97042CCFABEEDA701227529A12
radiusGroupName: gProfessors
sambaSID: S-1-0-0-28976

# User asac (group gLCFIB)
dn: uid=asac,ou=lcfib,ou=fib,dc=mydomain,dc=com
objectclass: radiusProfile
```

```
objectclass: radiusObjectProfile
objectclass: sambaSamAccount
uid: asac
cn: asac
sambaLmPassword:0x0F4DF6BD50F2B8B2193F552F6EA0491C
sambaNtPassword:0xDAE200A837B0819DDD7AD5FAD6C9B2F8
radiusGroupName: gLCFIB
sambaSID: S-1-0-0-28976
```

S'ha afegit l'*objectclass sambaSamAccount* de l'SCHEMA samba3.schema. *sambaSamAccount* ens permet utilitzar els atributs *sambaLmPassword* i *sambaNtPassword*. En aquest *objectclass*, l'atribut *sambaSID* és obligatori. Com que és un atribut que no utilitzarem, se li pot assignar qualsevol valor.

Per a poder incloure els usuaris en grups, s'utilitza l'atribut *radiusGroupName* del SCHEMA *RADIUS-LDAPv3.schema*. Per poder utilitzar aquest atribut és necessari incloure els *objectclass radiusProfile* i *radiusObjectProfile*.

Arranquem el *daemon* LDAP:

```
$>rcldap start
```

i carreguem fitxer *fib.ldif* a l'LDAP:

```
$>ldapadd -x -W -D "cn=admin,dc=mydomain,dc=com" -f fib.ldif
```

Ja hem configurat *openLDAP* i s'han afegit entrades a l'LDAP. Només queda configurar el mòdul LDAP de freeRADIUS.

4.4.1.2 Configuració LDAP en freeRADIUS

Editem la secció *authorize* del fitxer *etc/raddb/sites-available/inner-tunnel*. Concretament la que fe referencia a LDAP.

```
[…]
Authorization {
        […]
        #  The ldap module will set Auth-Type to LDAP if it has not already been set
        # Check if user must be authenticated in ActiveDirectory
        if (User-Name !~ /^AD[\\]{1,2}(.*)/i) {
                update control {
                        MS-CHAP-Use-NTLM-Auth := No
                }
                ldap
        }
        […]
}
[…]
```

AD és nom del controlador de domini d'Active Directory utilitzat per realitzar la prova pilot. Els dispositius que pertanyin a aquest domini enviaran el seu atribut *User-Name* en format *AD**username*.

Amb el codi en *unlang*:

```
(User-Name !~ /^AD[\\]{1,2}(.*)/i)
```

Es comprova que el nom d'usuari NO comenci per *AD*\, és a dir, que l'usuari s'ha d'autenticar en LDAP.

I amb el codi:

```
update control {
        MS-CHAP-Use-NTLM-Auth := No
        }
ldap
```

s'indica al servidor freeRADIUS que ha de inhabilitar el mòdul *MS-CHAP-Use-NTLM-Auth* (encarregat de fer l'autenticació via Active Directory) i ha autenticar en LDAP.

Si el nom d'usuari comença per *AD*\, el mòdul *MS-CHAP-Use-NTLM-Auth* quedarà habilitat i l'autenticació es realitzarà en Active Directory.

Per configurar el mòdul LDAP cal modificar el fitxer *etc/raddb/modules/ldap*:

```
ldap {
        #
        #  Note that this needs to match the name in the LDAP
        #  server certificate, if you're using ldaps.
        server = localhost
        identity = "cn=admin,dc=mydomain,dc=com"
        password = secret
        basedn = "ou=fib,dc=mydomain,dc=com"
        filter = "(uid=%{mschap:User-Name:-%{User-Name}})"
        […]
        #  Novell may require TLS encrypted sessions before returning
        #  the user's password.
        #
        # User's password stored using NTPassword
        password_attribute = sambaNtPassword
        […]
        #
        #  Group membership checking.  Disabled by default.
        #
        #
        groupname_attribute = cn
        groupmembership_filter = "(uid=%{mschap:User-Name:-%{User-Name}})"
        groupmembership_attribute = radiusGroupName
        […]
```

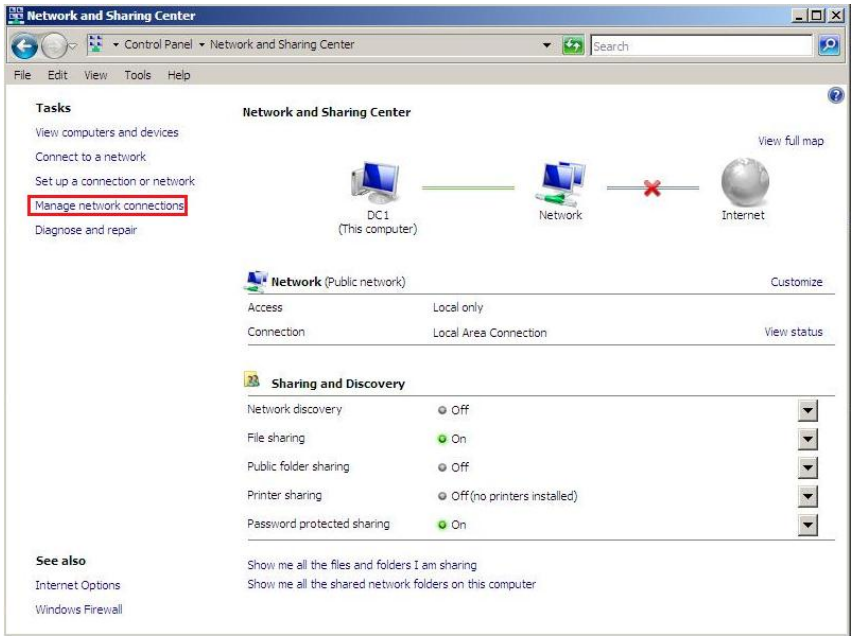
```
}
```

4.4.2 Active Directory

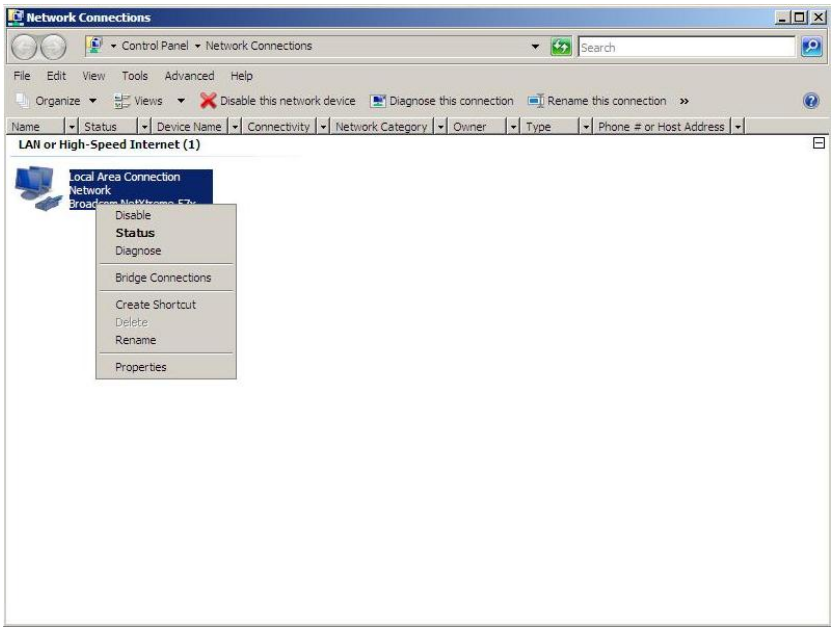
En aquest apartat s'expliquen els passos a seguir per a que freeRADIUS pugui autenticar els usuaris en Active Directory.

4.4.2.1 Instal·lació d'Active Directory

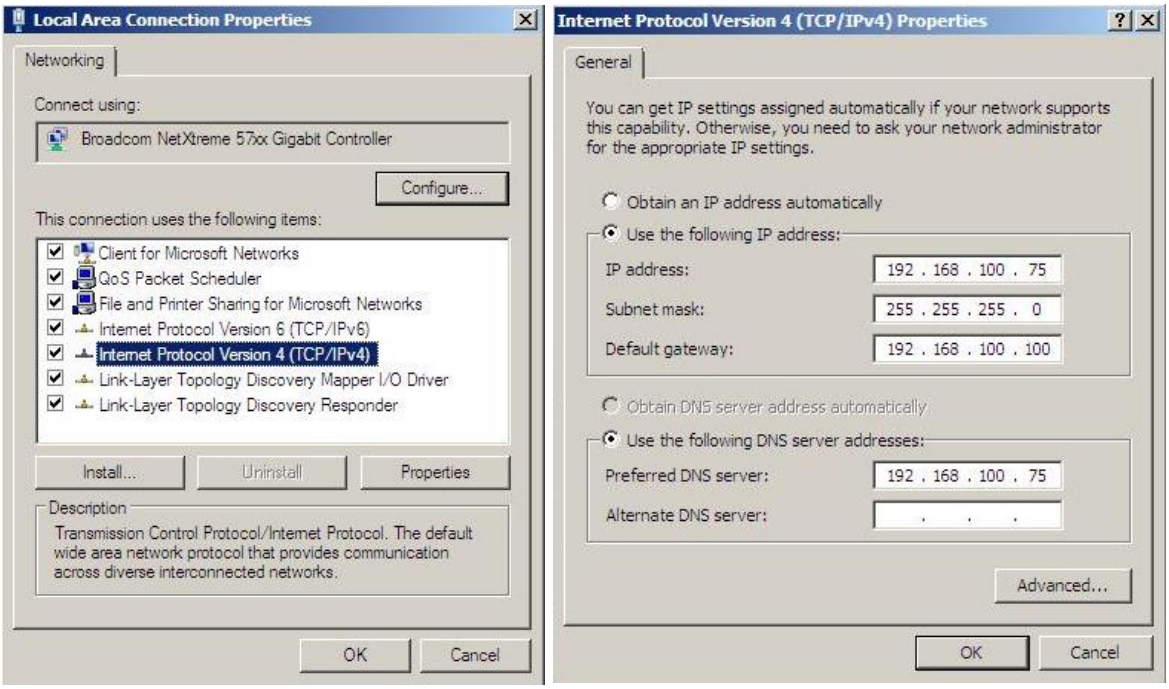
Dins del sistema Windows Server 2008 seleccionem "*Start -> Control Panel -> Network and Sharing Center -> Manage Network Connections*":



Seleccionem "*Local Area Connection Network -> Properties":*

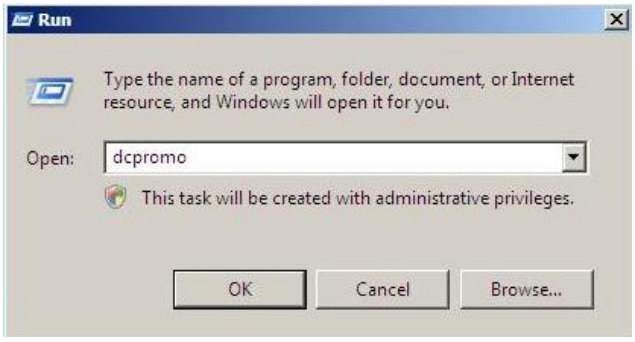


Escollim "*Internet Protocol Version 4 (TCP/IPv4) -> Properties"* i introduïm la configuració de xarxa. És important indicar que el "*Preferred DNS server"* sigui ell mateix:

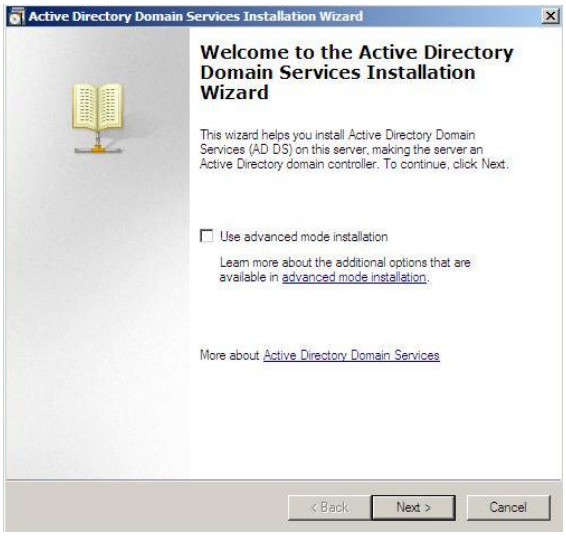


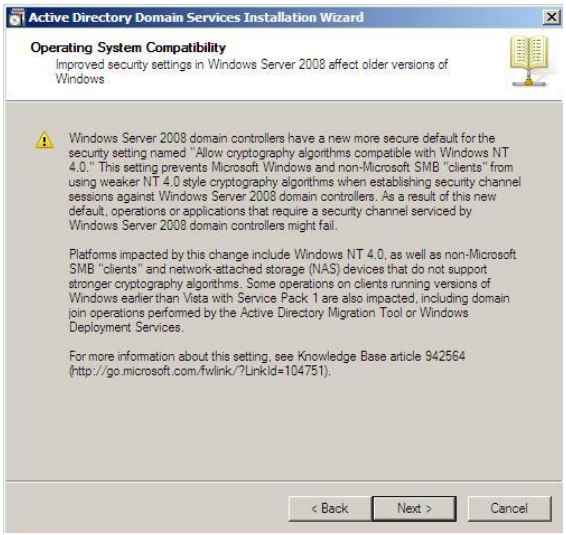
Un cop configurada la xarxa, ja podem instal·lar l'Active Directory:

Seleccionem "*Start -> Run"* i introduïm "*dcpromo"*:

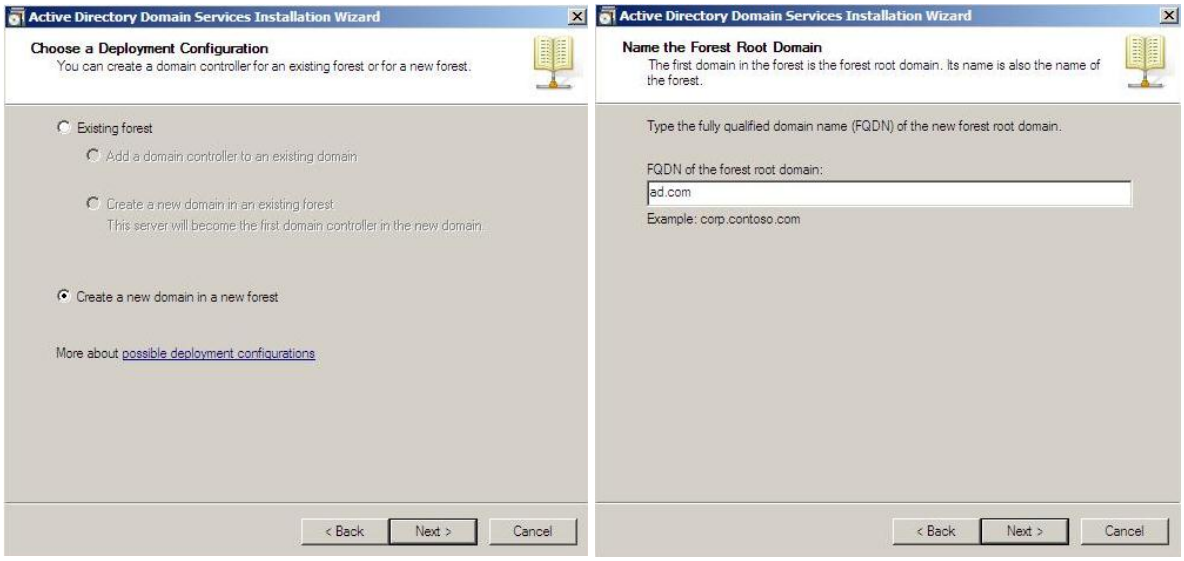


Començarà la instal·lació d'Active Direcotry. Cliquem "*Next"* en les dos primeres pantalles:

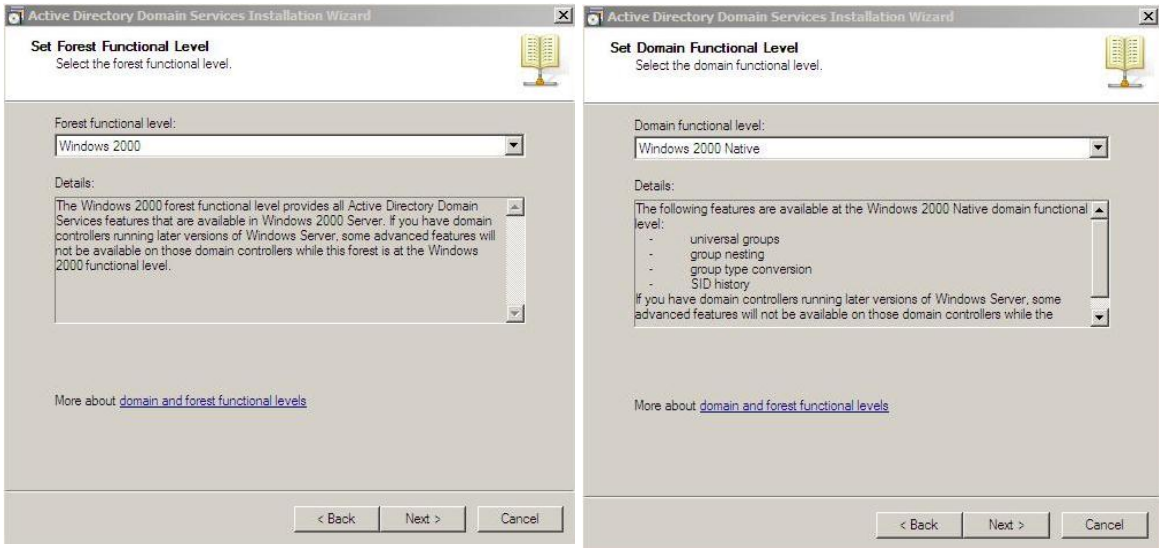




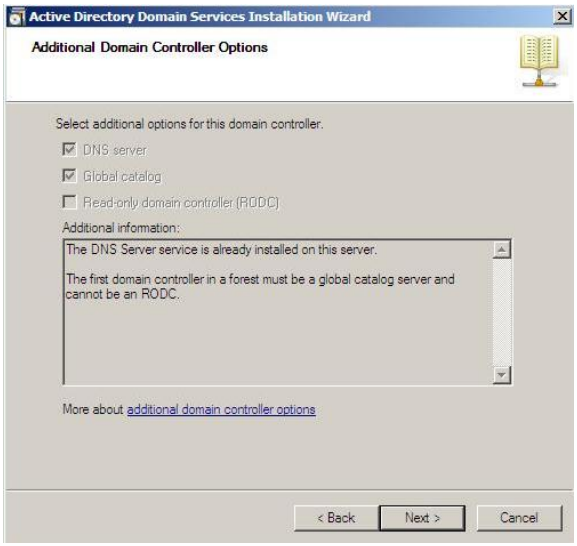
En la següent pantalla escollim "*Create a new domain in a new forest"* i introduïm el nom del nostre domini (ad.com):

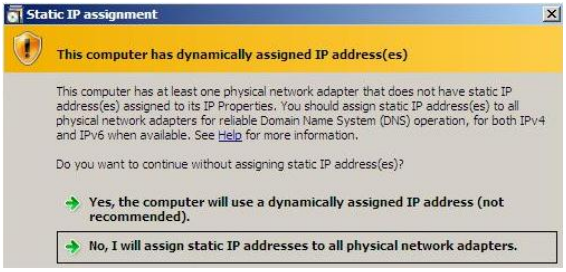


Seleccionem Windows 2000 com a "*Forest Functional Level"* i "*Domain Functional Level":*

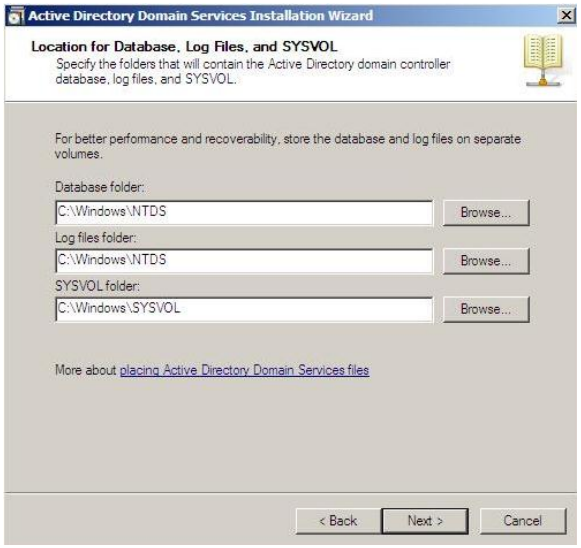


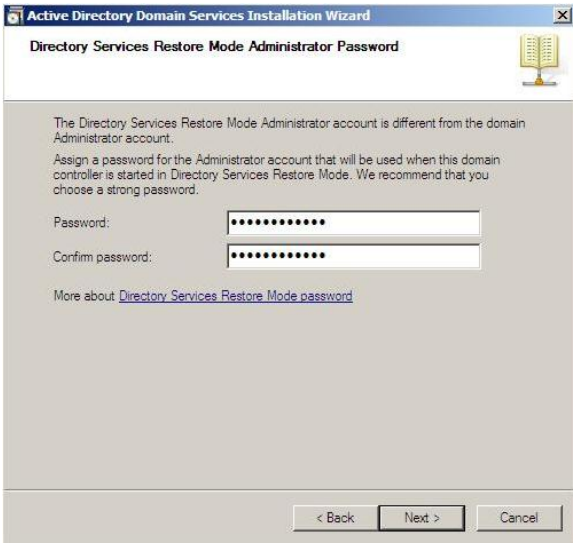
Acceptem la configuració per defecte en el menú "*Additional Controller Options"*. Quan aparegui la pantalla emergent seleccionem "*Yes"*:



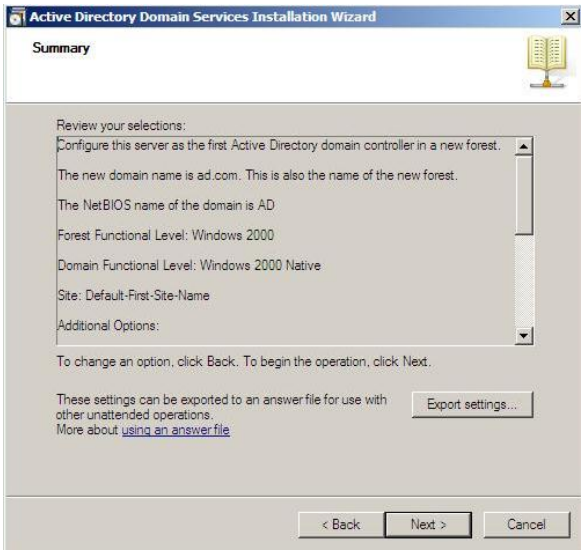


Acceptem les localitzacions de la base de dades, els fitxers de *log* i el SYSVOL per defecte i escollim un la contrasenya del controlador de domini:





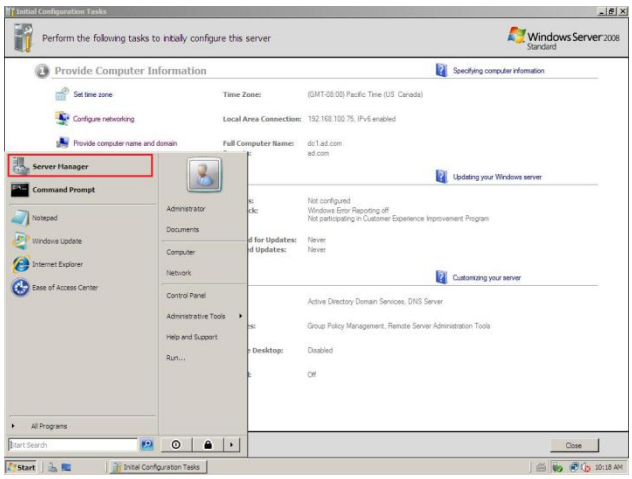
Comprovem la configuració escollida i iniciem la instal·lació del controlador de domini:



Per finalitzar la instal·lació simplement cal reiniciar el sistema.

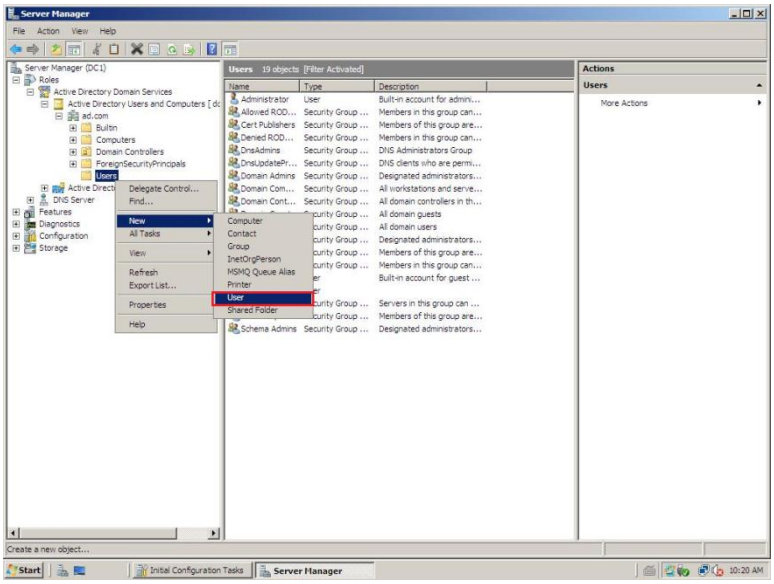
4.4.2.2 Afegir usuaris a l'Active Directory

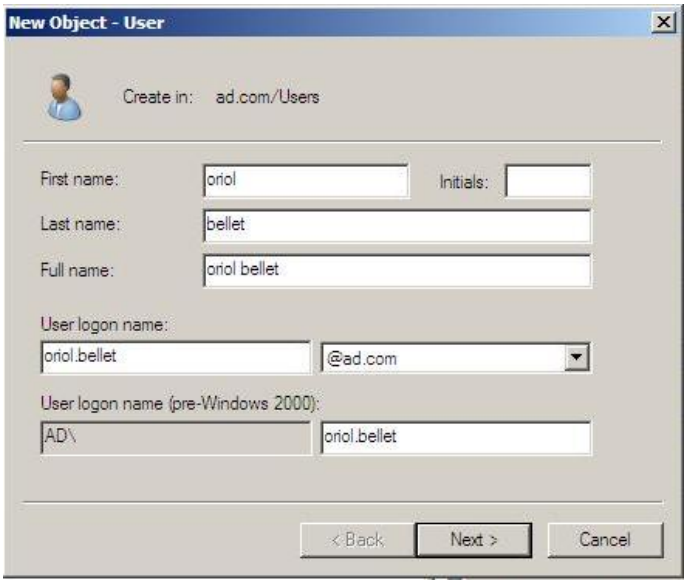
Per afegir un usuari a Active Directory seleccionem "*Start-> Server Manager*"



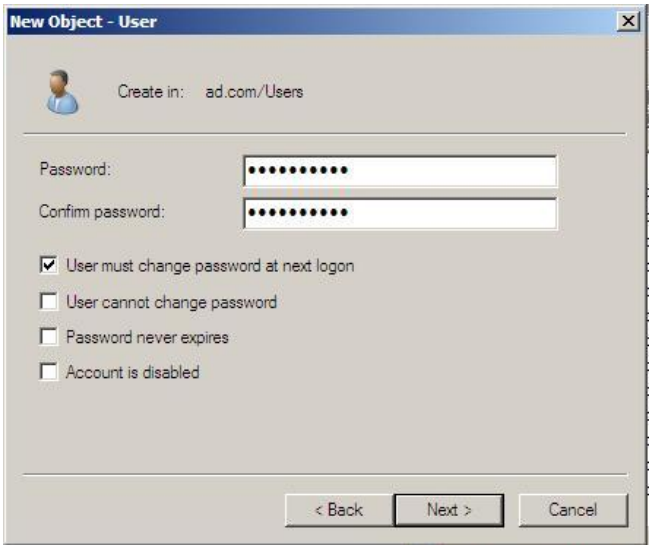
En el menú Server Manager seleccionem "*Roles -> Active Directory Domain Services -> Active Directory Users and Computers -> ad.com*".

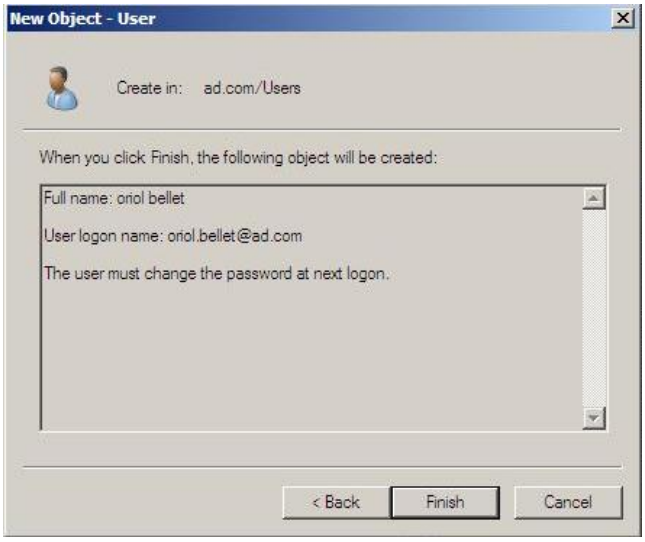
Amb el botó dret cliquem "*users*" i seleccionem "*new -> user*". En la pantalla emergent introduïm les dades del nou usuari:



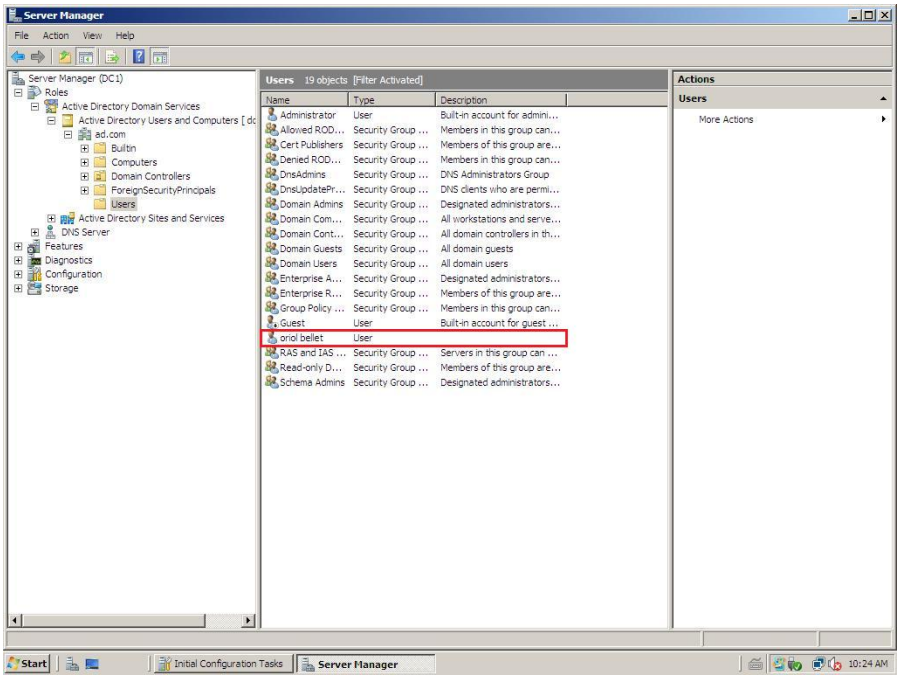


Finalment, li assignem una contrasenya inicial:





Finalment, comprovem que l'usuari creat apareix en el menú "*Users*":



4.4.2.2 Configurar el servidor per autenticar sobre Active Directory

A continuació s'explica com configurar freeRADIUS per a poder autenticar usuaris en Active Directory.

4.4.2.2.1 SAMBA

Per tal que el servidor freeRADIUS pugui autenticar els usuaris en Active Directory cal indicar a Samba el servidor i el domini. Editem el fitxer principal de configuració */etc/samba/smb.conf*:

```
# smb.conf is the main Samba configuration file. You find a full commented
# version at /usr/share/doc/packages/samba/examples/smb.conf.SUSE if the
# samba-doc package is installed.
# Date: 2009-10-27
[global]
        workgroup = AD
        security = ads
        realm = AD.COM
        password server = 192.168.100.75
```

```
        passdb backend = tdbsam
        printing = cups
        printcap name = cups
        printcap cache time = 750
        cups options = raw
        map to guest = Bad User
        include = /etc/samba/dhcp.conf
        logon path = \\%L\profiles\.msprofile
        logon home = \\%L\%U\.9xprofile
        logon drive = P:
 [homes]
[...]
```

4.4.2.2.3 Kerberos

També és necessari configurar *Kerberos*. Editem el fitxer */etc/krb5.conf* afegint la IP on es troba *l'Active Directory*:

```
[...]
[realms]
#       EXAMPLE.COM = {
#          kdc = kerberos.example.com
#                admin_server = kerberos.example.com
#       }

        AD.COM = {
                kdc = 192.168.100.75
        }
[...]
```

4.4.2.2.3 Afegir el servidor al domini de Windows

Per tal que freeRADIUS pugui autenticar usuaris a Active Directory caldrà que el servidor formi part del domini *ad.com*.

Arranquem els *daemons* de *Samba*:

```
$>smbd start
$>winbindd start
```

Executant la següent comanda (com a *root*) el servidor s'unirà al domini *ad.com*:

```
$>net ads join –U Administrator
Enter Administrator's password:
Joined 'hostname' to realm 'ad.com'
```

Administrator's password és la contrasenya de l'administrador del *Domain Controller.*

4.4.2.2.4 Proves d'autenticació

Per comprovar si la configuració s'ha fet correctament es pot executar la següent comanda, també com a *root*:

```
$>wbinfo –a oriol.bellet%oriolbelletADpassword
plaintext password authentication failed
Could not authenticate user oriol.bellet% oriolbelletADpassword with plaintext password
challenge/response password authentication succeeded
```

Si es rep com a resposta

challenge/response password authentication succeeded

significa que l'autenticació s'ha realitzat correctament.

La següent prova es pot realitzar utilitzant *ntlm_auth* (eina que utilitzarà freeRADIUS per autenticar usuaris en *l'Active Directory*):

```
$>/usr/bin/ntlm_auth --request-nt-key --domain=ad.com --username=oriol.bellet --password=
oriolbelletADpassword

NT_STATUS_OK: Success (0x0)
```

Si es rep com a resposta *Success* significarà que estem en condicions de configurar freeRADIUS per realitzar l'autenticació de forma automàtica.

4.4.2.3 Configuració Active Directory en freeRADIUS

Només cal indicar-li al servidor on pot trobar el binari *ntlm_auth* necessari per autenticar usuaris en Active Directory. Editem el fitxer */home/soft/freeradius/etc/modules/mschap*

```
mschap {
        […]
        # Windows sends us a username in the form of
        # DOMAIN\user, but sends the challenge response
        # based on only the user portion.  This hack
        # corrects for that incorrect behavior.
        #
        with_ntdomain_hack = yes

        […]
        # In that case, the mschap module will look at the User-Name
        # attribute, and do prefix/suffix checks in order to obtain
        # the "best" user name for the request.
        #
        ntlm_auth = "/usr/bin/ntlm_auth --request-nt-key --username=%{mschap:User-
        Name:-None} --domain=%{%{mschap:NT-Domain}:-ad.com} --
        challenge=%{mschap:Challenge:-00} --nt-response=%{mschap:NT-Response:-00}"
}
```

La configuració de la fase d'autenticació ja està finalitzada. Ara freeRADIUS és capaç de:

- Autenticar usuaris en LDAP.
- Autenticar els usuaris en Active Directory.
- Decidir sobre en quin servei de directori ha d'autenticar en cada cas.

4.5 Autorització

En aquest apartat es mostrarà com configurar el servidor per a poder realitzar la fase d'autorització.

4.5.1 MySQL

Abans de realitzar la configuració en freeRADIUS, caldrà crear la base de dades i totes les taules necessàries per realitzar l'autorització.

4.5.1.1 Configuració de l'SCHEMA SQL

Editem l'SCHEMA SQL que facilita freeRADIUS i que es troba a *etc/raddb/sql/mysql/schema.sql.* Canviem el nom de la taula *radgroupreply* per *authorize_boot_VLAN* i creem la taula *authorize_workstation_VLAN*:

```
#
# Table structure for table 'radgroupcheck'
#


CREATE TABLE radgroupcheck (
  id int(11) unsigned NOT NULL auto_increment,
  groupname varchar(64) NOT NULL default '',
  attribute varchar(64)  NOT NULL default '',
  op char(2) NOT NULL DEFAULT '==',
  value varchar(253)  NOT NULL default '',
  PRIMARY KEY  (id),
  KEY groupname (groupname(32))
) ;
```

```
#
# Table structure for table 'authorize_boot_VLAN'
#

CREATE TABLE authorize_boot_VLAN (
  id int(11) unsigned NOT NULL auto_increment,
  groupname varchar(64) NOT NULL default '',
  attribute varchar(64)  NOT NULL default '',
  op char(2) NOT NULL DEFAULT '=',
  value varchar(253)  NOT NULL default '',
  PRIMARY KEY  (id),
  KEY groupname (groupname(32))
) ;

#
# Table structure for table 'authorize_workstation_VLAN'
#

CREATE TABLE authorize_workstation_VLAN (
            id int(11) unsigned NOT NULL auto_increment,
            groupname varchar(64) NOT NULL default '',
            attribute varchar(64)  NOT NULL default '',
            op char(2) NOT NULL DEFAULT '=',
            value varchar(253)  NOT NULL default '',
            PRIMARY KEY  (id),
            KEY groupname (groupname(32))
) ;
[...]
```

4.5.1.2 Creació de la base de dades

Si s'utilitza una nova instal·lació de *MySQL*, primer caldrà crear les bases de dades d'administració. En el nostre cas, l'usuari *MySQL* en serà l'administrador:

```
$> mysql_install_db --user=mysql
```

Arrenquem el *daemon MySQL* en mode *safe*:

```
$> sudo -u mysql mysqld_safe
```

Creem la base de dades i usuari *radius* (que tindrà tots els permisos sobre la nova base de dades).

```
$>mysqladmin -u root password $PASSWORD

mysql> CREATE DATABASE radius;

mysql> CREATE USER 'radius@localhost';

mysql> GRANT ALL PRIVILEGES ON radius.* TO 'radius'@'localhost' IDENTIFIED BY 'radpass'
WITH GRANT OPTION;

mysql> exit;
```

Creem les taules de la BD mitjançant l'*SCHEMA* SQL anteriorment editat:

```
$>mysql -uroot -p radius < /home/soft/freeradius/etc/raddb/sql/mysql/schema.sql
```

Per accedir a la base de dades *radius*:

```
$>mysql –uradius –p

mysql>use radius;
```

4.5.1.3 Polítiques d'autorització a MySQL

En aquest apartat es mostra com emmagatzemar en la base de dades la informació relacionada amb l'autorització. Aquestes dades són:

- Assignar un perfil a les estacions de treball.
- Definir les polítiques d'autorització per a cada perfil.

4.5.1.3.1 Assignar un perfil a les estacions de treball

Per assignar un perfil a una estació de treball creem una nova entrada a la taula *radusergroup*:

```
mysql> insert into radusergroup values ('00-1B-21-04-C2-50','A5',1);
```

En l'exemple anterior s'ha assignat el perfil *A5* a l'estació de treball amb adreça MAC *00-1B-21-04-C2-50*. Caldrà assignar un perfil a totes les estacions de treball del sistema.

4.5.1.3.2 Definir les polítiques d'autorització per a cada perfil

Per assignar una VLAN a un perfil cal crear noves entrades a les taules *authorize_boot_VLAN* i *authorize_workstation_VLAN*:

```
mysql> insert into authorize_boot_VLAN  values (NULL,'A5','Tunnel-Type','=','VLAN');

mysql> insert into authorize_boot_VLAN  values (NULL,'A5','Tunnel-Medium-Type','=','IEEE-802');

mysql> insert into authorize_boot_VLAN  values (NULL,'A5','Tunnel-Private-Group-Id','=','A5_BOOT_VLAN');

mysql> insert into authorize_workstation_VLAN values (NULL,'A5','Tunnel-Type','=','VLAN');

mysql> insert into authorize_workstation_VLAN values (NULL,'A5','Tunnel-Medium-Type','=','IEEE-802');

mysql> insert into authorize_workstation_VLAN values (NULL,'A5','Tunnel-Private-Group-Id','=', A5_VLAN');
```

En l'exemple anterior les estacions de treball amb perfil *A5* seran col·locades a *l'VLAN A5_BOOT_VLAN* per poder arrencar per xarxa i a l'*A5_VLAN* quan realitzin l'autenticació 802.1x.

Caldrà repetir aquest procés per a cada perfil d'estacions de treball existent.

4.5.2 Configuració SQL en freeRADIUS

En aquest apartat es mostra com configurar freeRADIUS per a que pugui autoritzar els usuaris utilitzant la base de dades creada.

4.5.2.1 Creació dels mòduls SQL

Els mòduls SQL de freeRADIUS són gestionats a través del fitxer *etc/raddb/sql.conf*. En la configuració per defecte només s'utilitza un únic mòdul ja que sempre es realitzen les mateixes consultes SQL sobre les mateixes taules.

En el nostre cas necessitarem varis mòduls SQL. Si volem autoritzar una estació de treball per a que pugui arrencar per xarxa, hem de consultar la taula d'autoritzacions *authorize_boot_VLAN*. En canvi, si volem una autorització global d'una estació de treball hem de consultar la taula *authorize_workstation_VLAN*. Per tant, les taules a consultar i les sentències SQL variaran depenent del tipus d'autorització a realitzar.

S'ha creat un mòdul SQL per a cada tipus d'autorització en el fitxer *sql.conf*. El mòdul *is_authorized_to_boot_from_network* estarà associat a la taula *authorize_boot_VLAN*, i el mòdul *is_workstation* estarà associat a la taula *authorize_workstation_VLAN*.

```
# Configuration for the SQL module
#
# The database schemas and queries are located in subdirectories:
#
#       sql/DB/schema.sql       Schema
#       sql/DB/dialup.conf      Basic dialup (including policy) queries
#       sql/DB/counter.conf     counter
#       sql/DB/ippool.conf      IP Pools in SQL
#       sql/DB/ippool.sql       schema for IP pools.
#
# Where "DB" is mysql, mssql, oracle, or postgresql.
#


sql is_authorized_to_boot_from_network {
```

```
        #
        #  Set the database to one of:
        #
        #          mysql, mssql, oracle, postgresql
        #
        database = "mysql"

[...]

        # If you want both stop and start records logged to the
        # same SQL table, leave this as is.  If you want them in
        # different tables, put the start table in acct_table1
        # and stop table in acct_table2
        #acct_table1 = "radacct"
        #acct_table2 = "radacct"

        # Allow for storing data after authentication
        #postauth_table = "radpostauth"

        authcheck_table = "radcheck"
        authreply_table = "radreply"

        groupcheck_table = "radgroupcheck"
        groupreply_table = "authorize_boot_VLAN"

        # Table to keep group info
        usergroup_table = "radusergroup"

[...]
```

```
        $INCLUDE sql/${database}/dialup-auth.conf


}


sql is_workstation {
        database = "mysql"


        driver = "rlm_sql_${database}"


        server = "localhost"


        login = "radius"
        password = "radpass"


        radius_db = "radius"


        authcheck_table = "radcheck"
        authreply_table = "radreply"


        groupcheck_table = "radgroupcheck"
        groupreply_table = "authorize_workstation_VLAN"


        usergroup_table = "radusergroup"


        deletestalesessions = yes


        sqltrace = no
        sqltracefile = ${logdir}/sqltrace.sql
```

```
        num_sql_socks = 5

        connect_failure_retry_delay = 60

        lifetime = 0

        max_queries = 0

        nas_table = "nas"

        $INCLUDE sql/${database}/dialup-auth.conf
}
```

Més endavant es crearà un altre mòdul SQL per gestionar el procés *d'accountig*. Per aquest motiu, les línies on es referencien les taules *d'accounting* han estat comentades en aquests mòduls dedicats a la fase d'autorització.

La variable $INCLUDE de cada mòdul fa referència al fitxer on es troben les sentències SQL. Per defecte es troben en el fitxer *etc/raddb/sql/mysql/dialup.conf*. En la nostra implementació s'ha dividit aquest fitxer en dos per tal de separar la part d'autenticació/autorització de la part *d'accounting*:

- ***dialup-auth.conf*** Dedicat a les consultes SQL d'autenticació/autorització.
- ***dialup-acct.conf*** Dedicat a les consultes SQL *d'accounting.*

Copiem les consultes SQL del fitxer *dialup.conf* relaciones amb l'autenticació/autorització al nou fitxer *etc/raddb/sql/mysql/dialup-auth.conf:*

```
sql_user_name = "%{User-Name}"

nas_query = "SELECT id, nasname, shortname, type, secret, server FROM ${nas_table}"

authorize_check_query = "SELECT id, username, attribute, value, op \
FROM ${authcheck_table} \
```

```
WHERE username = '%{SQL-User-Name}' \
ORDER BY id"

authorize_reply_query = "SELECT id, username, attribute, value, op \
FROM ${authreply_table} \
WHERE username = '%{SQL-User-Name}' \
ORDER BY id"

group_membership_query = "SELECT groupname \
FROM ${usergroup_table} \
WHERE username = '%{Calling-Station-Id}' \
ORDER BY priority"

authorize_group_check_query = "SELECT id, groupname, attribute, \
Value, op \
FROM ${groupcheck_table} \
WHERE groupname = '%{Sql-Group}' \
ORDER BY id"

authorize_group_reply_query = "SELECT id, groupname, attribute, \
value, op \
FROM ${groupreply_table} \
WHERE groupname = '%{Sql-Group}' \
ORDER BY id"
```

Cal notar que en la consulta sobre la taula *usergroupreply* es comprova que el *username* sigui igual a l'adreça MAC del dispositiu i no al nom d'usuari.

4.5.2.2 Autoritzacions d'arrancada per a les estacions de treball

A continuació es descriu com configurar les polítiques d'autorització necessàries per a que les estacions de treball puguin arrencar per xarxa.

Les estacions de treball realitzen l'autenticació MAB per a poder tenir permisos per arrancar per xarxa.

El format del camp *Calling-Station-Id* de les trames RADIUS, que conté l'adreça MAC del *Supplicant*, no està estandarditzat. La conseqüència d'això és que cada fabricant *d'Authenticators* (client RADIUS) és lliure d'enviar l'adreça MAC del *Supplicant* utilitzant el format que vulgui.

Per tal d'homogeneïtzar el sistema, s'ha decidit estandarditzar aquest procés. Crearem un mòdul freeRADIUS anomenat *rewrite_calling_station_id* que s'encarregarà de sobreescriure el paràmetre *Calling-Station-Id* per deixar-lo en un format estàndard (AA-BB-CC-DD-EE-FF).

Definim el nou mòdul al fitxer *etc/raddb/policy.conf:*

```
[…]
#
# Rewrite called station id attribute into a standard format.
#
rewrite_calling_station_id {
    if (Calling-Station-Id =~ /([0-9a-f]{2})[-:]?([0-9a-f]{2})[-:]?([0-9a-f]{2})[-:]?([0-9a-f]{2})[-:]?([0-9a-f]{2})[-:]?([0-9a-f]{2})/i){
        update request {
            Calling-Station-Id := "%{tolower:%{1}-%{2}-%{3}-%{4}-%{5}-%{6}}"
        }
    }
    else {
        noop
    }
}
```

Afegim la crida al nou mòdul en el fitxer *etc/raddb/sites-available/default* :

```
[...]
authorize {
        #
        #  The preprocess module takes care of sanitizing some bizarre
        #  attributes in the request, and turning them into attributes
        #  which are more standard.
        #
        #  It takes care of processing the 'raddb/hints' and the
        #  'raddb/huntgroups' files.
        preprocess

        # Cleaning up the Calling-Station-Id
```

```
        rewrite_calling_station_id
[...]
```

Un cop estandarditzat el paràmetre *Calling-Station-Id,* ja es pot realitzar l'autenticació MAB. Afegim noves línies al fitxer *etc/raddb/sites-available/default:*

```
[...]
        #  It takes care of processing the 'raddb/hints' and the
        #  'raddb/huntgroups' files.
        preprocess


        # Cleaning up the Calling-Station-Id
        rewrite_calling_station_id


        #Check if received package is a EAP-Message (mac-auth is not encapsulated in EAP-
        #Message)
        if (!EAP-Message) {
                #mac-auth
                is_authorized_to_boot_from_network
                if (ok) {
                        #If successful mac-auth, accept authentication request
                        update control {
                                Auth-Type := Accept
                        }
                }
        }
[...]
```

A diferència de 802.1x, les trames MAB no estan encapsulades en una trama EAP. La línia

if (!EAP-Message) {

ens ajuda a diferenciar si la trama rebuda pel servidor és 802.1x o MAB.

Si és MAB, cridarem al mòdul SQL *is_allowed_boot_from_network* que retornarà *ok* en cas que l'estació de treball tingui permisos per utilitzar recursos de xarxa en temps d'arrencada.

A l'interpretar la línia:

update control {

Auth-Type := Accept

}

el servidor freeRADIUS envia una trama *Access-Accept* al commutador afegint-li les polítiques d'autorització definides a la taula *is_allowed_boot_from_network* de la base de dades.

4.5.2.3 Autoritzacions globals de les estacions de treball

A continuació es descriu com configurar les polítiques d'autorització globals de les estacions de treball. Afegim les següents línies al fitxer *etc/raddb/sites-available/default:*

```
[...]


authorize {
        #
        #  The preprocess module takes care of sanitizing some bizarre
        #  attributes in the request, and turning them into attributes
        #  which are more standard.
        #
        #  It takes care of processing the 'raddb/hints' and the
        #  'raddb/huntgroups' files.
        preprocess
```

```
        # if cleaning up the Calling-Station-Id...
        rewrite_calling_station_id


        if (!EAP-Message) {
                #Comprovem la BD
                comprovar_macs
                if (ok) {
                        update control {
                                Auth-Type := Accept
                        }
                }
        }


        #Check if the Calling-Station-Id of the Supplicant is a workstations. If it is,
        #authorization is done through mysql
        is_workstation
        if (!ok) {
                #If not, authorization is done by users in users file
                files
        }
[...]
```

En cas que la petició es realitzi des d'una estació de treball, el mòdul *is_workstation* serà qui definirà la política d'autorització i retornarà *ok*.

Si la petició es realitza des d'un dispositiu extern, es realitzarà l'autorització per usuari, gestionada pel mòdul freeRADIUS *files.*

El mòdul *files* només s'executarà en certes circumstàncies (quan l'usuari no estigui connectat en una estació de treball). Per tant, cal comentar la línia del fitxer *default* que fa que el mòdul files s'executi sempre:

```
[...]
        #
        #  Read the 'users' file
        #files
[...]
```

4.5.2.4 Autorització d'usuaris

Les polítiques d'autorització basades en grups d'usuaris estan definides en el fitxer *etc/raddb/users*. El mòdul *files* és l'encarregat d'interpretar aquest fitxer.

Per definir els grups d'usuaris, editem el fitxer *users*:

```
[…]


#
#       For a list of RADIUS attributes, and links to their definitions,
#       see:
#
#       http://www.freeradius.org/rfc/attributes.html
#


DEFAULT Ldap-Group == gLCFIB
        Tunnel-Type = VLAN,
        Tunnel-Medium-Type = IEEE-802,
        Tunnel-Private-Group-Id = "LCFIB_VLAN"


DEFAULT Ldap-Group == gStudents
        Tunnel-Type = VLAN,
        Tunnel-Medium-Type = IEEE-802,
        Tunnel-Private-Group-Id = "STUDENTS_VLAN"
```

```
DEFAULT Ldap-Group == gProfessors
        Tunnel-Type = VLAN,
        Tunnel-Medium-Type = IEEE-802,
        Tunnel-Private-Group-Id = "PROFESSORS_VLAN"

[…]
```

Amb la línia

DEFAULT Ldap-Group == X

es comprova si l'usuari que està realitzant la petició d'autorització pertany al grup X (l'assignació al grup es realitza quan l'usuari es dóna d'altra a l'LDAP).

Si hi pertany, se li aplicarà la política d'autorització del grup. Si no, es comprovarà si forma part del següent grup.

En aquest punt freeRADIUS ja està preparat per realitzar les fases d'autenticació i autorització.

4.6 Accounting

4.6.1 Configuració de l'accounting en freeRADIUS

S'ha creat un mòdul SQL per gestionar l'emmagatzemament de les dades *d'accounting*. Afegim el nou mòdul a continuació dels existents en el fitxer *etc/raddb/sql.conf:*

```
sql sql_accounting{
        database = "mysql"
        driver = "rlm_sql_${database}"
        server = "localhost"
        login = "radius"
        password = "radpass"
        radius_db = "radius"
```

```
        acct_table1 = "radacct"
        acct_table2 = "radacct"

        deletestalesessions = yes

        sqltracefile = ${logdir}/sqltrace.sql

        num_sql_socks = 5

        connect_failure_retry_delay = 60
        lifetime = 0

        max_queries = 0

        nas_table = "nas"

        $INCLUDE sql/${database}/dialup-acct.conf
}
```

Notar que en aquest mòdul no s'han assignat les taules dedicades a l'autenticació/autorització. ja que seran tractades pels altres mòduls SQL.

Creem el fitxer de sentències SQL *d'accounting* associat al nou mòdul *etc/raddb/sql/mysql/dialup-acct.conf:*

```
sql_user_name = "%{User-Name}"


nas_query = "SELECT id, nasname, shortname, type, secret, server FROM ${nas_table}"


accounting_onoff_query = "\
        UPDATE ${acct_table1} \
        SET \
        acctstoptime      =  '%S', \
        acctsessiontime   =  unix_timestamp('%S') - \
                unix_timestamp(acctstarttime), \
        acctterminatecause =  '%{Acct-Terminate-Cause}', \
        acctstopdelay     =  %{%{Acct-Delay-Time}:-0} \
        WHERE acctstoptime IS NULL \
        AND nasipaddress     =  '%{NAS-IP-Address}' \
        AND acctstarttime    <= '%S'"

accounting_update_query = " \
        UPDATE ${acct_table1} \
        SET \
        framedipaddress = '%{Framed-IP-Address}', \
        acctsessiontime    = '%{Acct-Session-Time}', \
```

```
        acctinputoctets    = '%{%{Acct-Input-Gigawords}:-0}'  << 32 | \
                '%{%{Acct-Input-Octets}:-0}', \
        acctoutputoctets   = '%{%{Acct-Output-Gigawords}:-0}' << 32 | \
                '%{%{Acct-Output-Octets}:-0}' \
        WHERE acctsessionid = '%{Acct-Session-Id}' \
        AND username       = '%{SQL-User-Name}' \
        AND nasipaddress   = '%{NAS-IP-Address}'"

accounting_update_query_alt = " \
        INSERT INTO ${acct_table1} \
         (acctsessionid,   acctuniqueid,    username, \
        realm,         nasipaddress,    nasportid, \
        nasporttype,    acctstarttime,    acctsessiontime, \
        acctauthentic,   connectinfo_start, acctinputoctets, \
        acctoutputoctets, calledstationid,  callingstationid, \
        servicetype,    framedprotocol,   framedipaddress, \
        acctstartdelay,  xascendsessionsvrkey) \
        VALUES \
         ('%{Acct-Session-Id}', '%{Acct-Unique-Session-Id}', \
        '%{SQL-User-Name}', \
        '%{Realm}', '%{NAS-IP-Address}', '%{NAS-Port}', \
        '%{NAS-Port-Type}', \
        DATE_SUB('%S', \
                INTERVAL (%{%{Acct-Session-Time}:-0} + \
                        %{%{Acct-Delay-Time}:-0}) SECOND), \
                '%{Acct-Session-Time}', \
        '%{Acct-Authentic}', '', \
        '%{%{Acct-Input-Gigawords}:-0}' << 32 | \
        '%{%{Acct-Input-Octets}:-0}', \
        '%{%{Acct-Output-Gigawords}:-0}' << 32 | \
        '%{%{Acct-Output-Octets}:-0}', \
        '%{Called-Station-Id}', '%{Calling-Station-Id}', \
        '%{Service-Type}', '%{Framed-Protocol}', \
        '%{Framed-IP-Address}', \
        '0', '%{X-Ascend-Session-Svr-Key}')"

accounting_start_query = " \
        INSERT INTO ${acct_table1} \
                 (acctsessionid,   acctuniqueid,    username, \
                realm,         nasipaddress,    nasportid, \
                nasporttype,    acctstarttime,    acctstoptime, \
                acctsessiontime,  acctauthentic,   connectinfo_start, \
                connectinfo_stop, acctinputoctets,  acctoutputoctets, \
                calledstationid,  callingstationid, acctterminatecause, \
                servicetype,    framedprotocol,   framedipaddress, \
                acctstartdelay,   acctstopdelay,   xascendsessionsvrkey) \
        VALUES \
                 ('%{Acct-Session-Id}', '%{Acct-Unique-Session-Id}', \
                '%{SQL-User-Name}', \
                '%{Realm}', '%{NAS-IP-Address}', '%{NAS-Port}', \
                '%{NAS-Port-Type}', '%S', NULL, \
                '0', '%{Acct-Authentic}', '%{Connect-Info}', \
```

```
                '', '0', '0', \
                '%{Called-Station-Id}', '%{Calling-Station-Id}', '', \
                '%{Service-Type}', '%{Framed-Protocol}', '%{Framed-IP-Address}', \
                '%{%{Acct-Delay-Time}:-0}', '0', '%{X-Ascend-Session-Svr-Key}')"

accounting_start_query_alt  = " \
        UPDATE ${acct_table1} SET \
                acctstarttime    = '%S', \
                acctstartdelay   = '%{%{Acct-Delay-Time}:-0}', \
                connectinfo_start = '%{Connect-Info}' \
        WHERE acctsessionid  = '%{Acct-Session-Id}' \
        AND username        = '%{SQL-User-Name}' \
        AND nasipaddress    = '%{NAS-IP-Address}'"

accounting_stop_query = " \
        UPDATE ${acct_table2} SET \
                acctstoptime     = '%S', \
                acctsessiontime   = '%{Acct-Session-Time}', \
                acctinputoctets   = '%{%{Acct-Input-Gigawords}:-0}' << 32 | \
                        '%{%{Acct-Input-Octets}:-0}', \
                acctoutputoctets  = '%{%{Acct-Output-Gigawords}:-0}' << 32 | \
                        '%{%{Acct-Output-Octets}:-0}', \
                acctterminatecause = '%{Acct-Terminate-Cause}', \
                acctstopdelay     = '%{%{Acct-Delay-Time}:-0}', \
                connectinfo_stop  = '%{Connect-Info}' \
        WHERE acctsessionid  = '%{Acct-Session-Id}' \
        AND username        = '%{SQL-User-Name}' \
        AND nasipaddress    = '%{NAS-IP-Address}'"

accounting_stop_query_alt = " \
        INSERT INTO ${acct_table2} \
                (acctsessionid, acctuniqueid, username, \
                realm, nasipaddress, nasportid, \
                nasporttype, acctstarttime, acctstoptime, \
                acctsessiontime, acctauthentic, connectinfo_start, \
                connectinfo_stop, acctinputoctets, acctoutputoctets, \
                calledstationid, callingstationid, acctterminatecause, \
                servicetype, framedprotocol, framedipaddress, \
                acctstartdelay, acctstopdelay) \
        VALUES \
                ('%{Acct-Session-Id}', '%{Acct-Unique-Session-Id}', \
                '%{SQL-User-Name}', \
                '%{Realm}', '%{NAS-IP-Address}', '%{NAS-Port}', \
                '%{NAS-Port-Type}', \
        DATE_SUB('%S', \
                INTERVAL (%{%{Acct-Session-Time}:-0} + \
                %{%{Acct-Delay-Time}:-0}) SECOND), \
        '%S', '%{Acct-Session-Time}', '%{Acct-Authentic}', '', \
        '%{Connect-Info}', \
        '%{%{Acct-Input-Gigawords}:-0}' << 32 | \
        '%{%{Acct-Input-Octets}:-0}', \
        '%{%{Acct-Output-Gigawords}:-0}' << 32 | \
```

```
        '%{%{Acct-Output-Octets}:-0}', \
        '%{Called-Station-Id}', '%{Calling-Station-Id}', \
        '%{Acct-Terminate-Cause}', \
        '%{Service-Type}', '%{Framed-Protocol}', '%{Framed-IP-Address}', \
        '0', '%{%{Acct-Delay-Time}:-0}')"
```

Un cop creat el mòdul SQL, afegim la crida en l'apartat *accounting* del fitxer *etc/raddb/sites-available/default*

```
#
#  Accounting.  Log the accounting data.
#
accounting {
[…]
        #
        #  Log traffic to an SQL database.
        #
        #  See "Accounting queries" in sql.conf
#       sql
        sql_accounting
[…]
}
```

freeRADIUS ja és capaç de realitzar les fases d'autenticació, autorització i Accounting.

4.7 Els *Supplicants*

En aquest apartat es mostra com configurar els *Supplicants* per realitzar el procés d'autenticació. En els *Supplicant* de Microsoft Windows quatre tasques són necessàries:

- Habilitar l'autenticació 802.1x
- Instal·lar certificat de la CA.
- Configurar l'autenticació 802.1x.
- Afegir el Supplicant a un domini Windows.

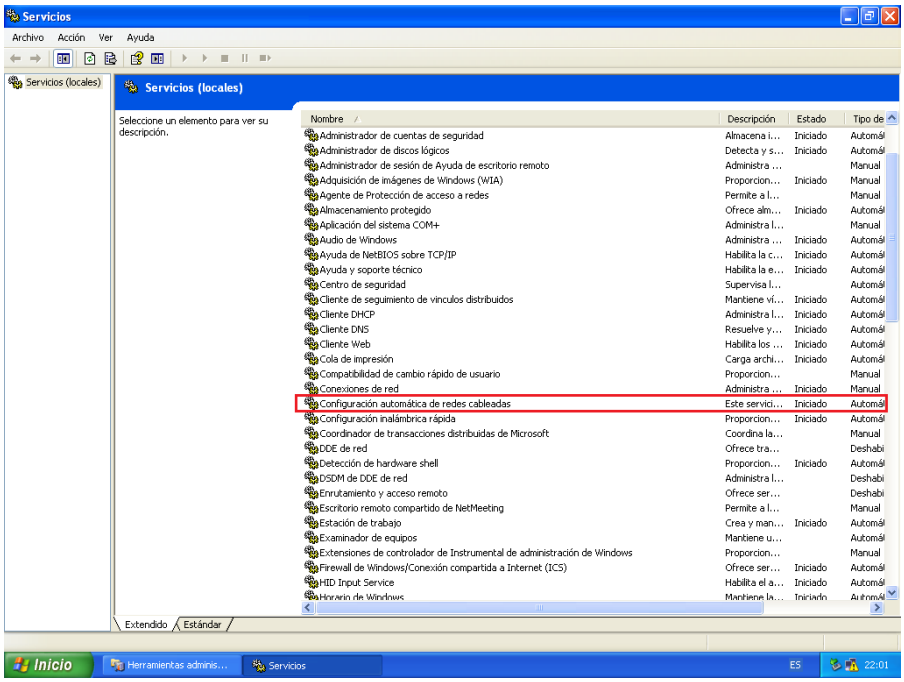
En els *Supplicants* Linux s'explicaran les següents accions:

- Instal·lar certificat CA.
- Configurar l'autenticació 802.1x

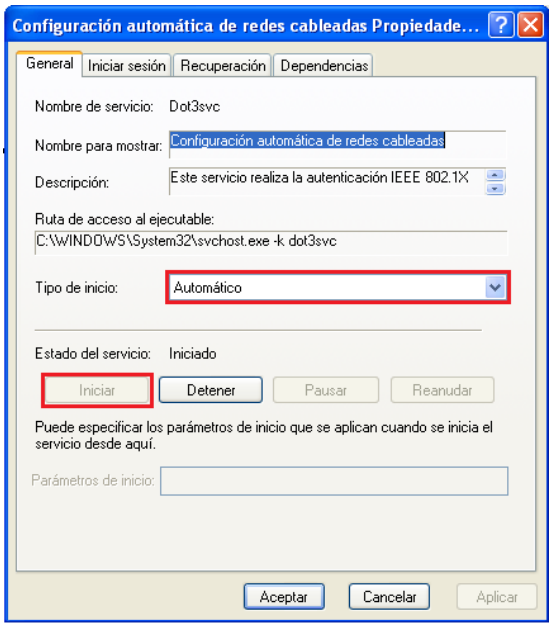
4.7.1 Windows XP

4.7.1.1 Habilitar l'autenticació 802.1x en Windows XP

"Inicio -> Panel de control -> Herramientas Administrativas -> Servicios" i seleccionem "*Configuración automática de redes cableadas":*

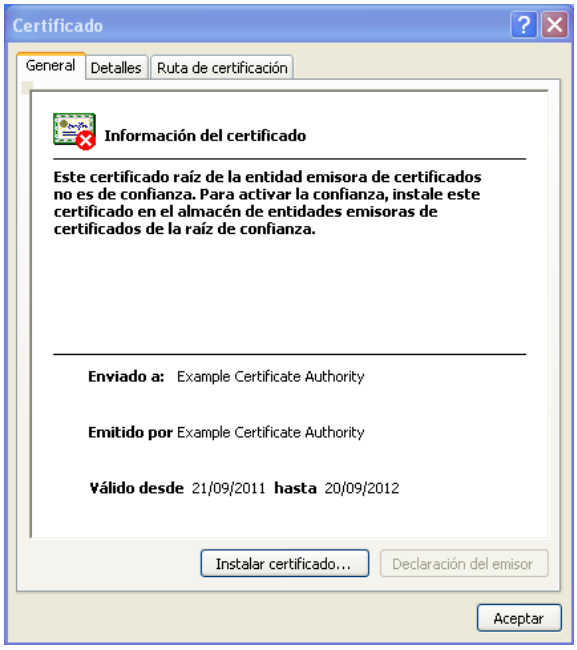


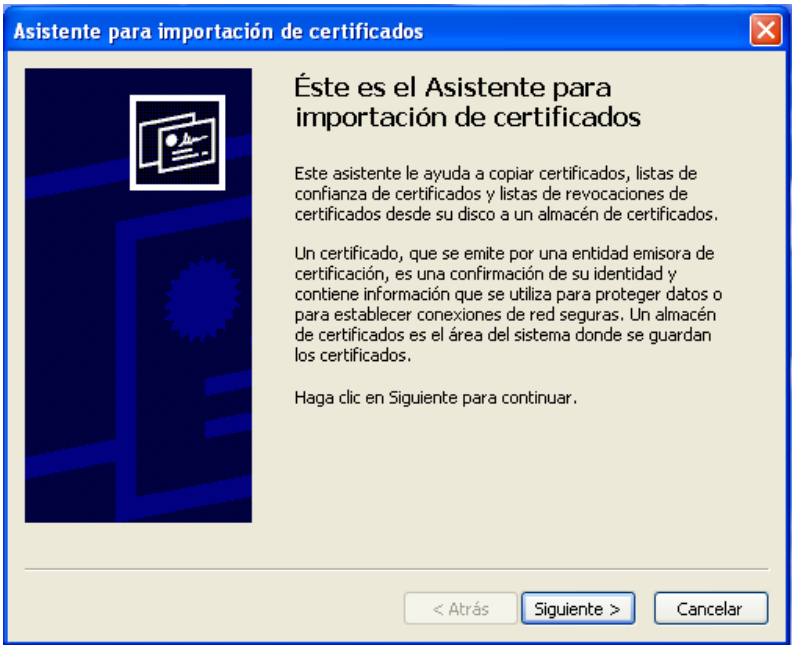
Com a "*Tipo de inicio"* escollim "*Automático"* i arranquem el servei mitjançant el botó "*Iniciar":*



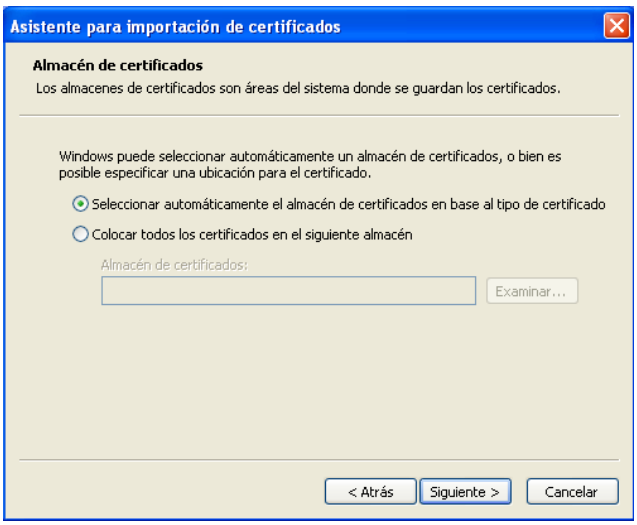
4.7.1.2 Afegir el certificat de l'Autoritat Certificadora en Windows XP

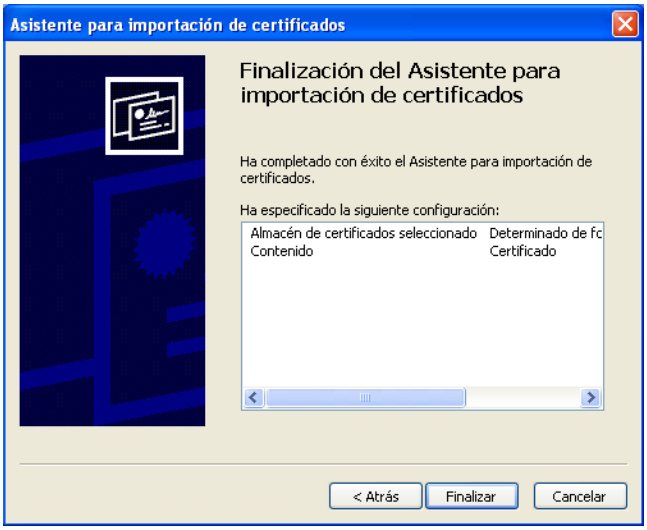
Obrim el certificat per tal d'incorporar-lo al sistema. Apareixerà una finestra on podrem veure les seves dades. Un cop comprovades seleccionem "*Instalar certificado…*" i en la pantalla emergent "*Siguiente*":



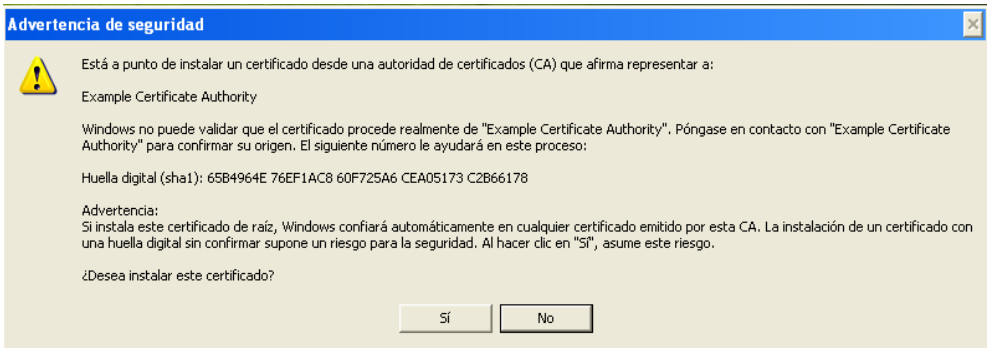


El següent menú ens preguntarà en quin *magatzem de certificats* volem desar el nou certificat. Escollim l'opció per defecte i finalitzem la instal·lació del certificat:



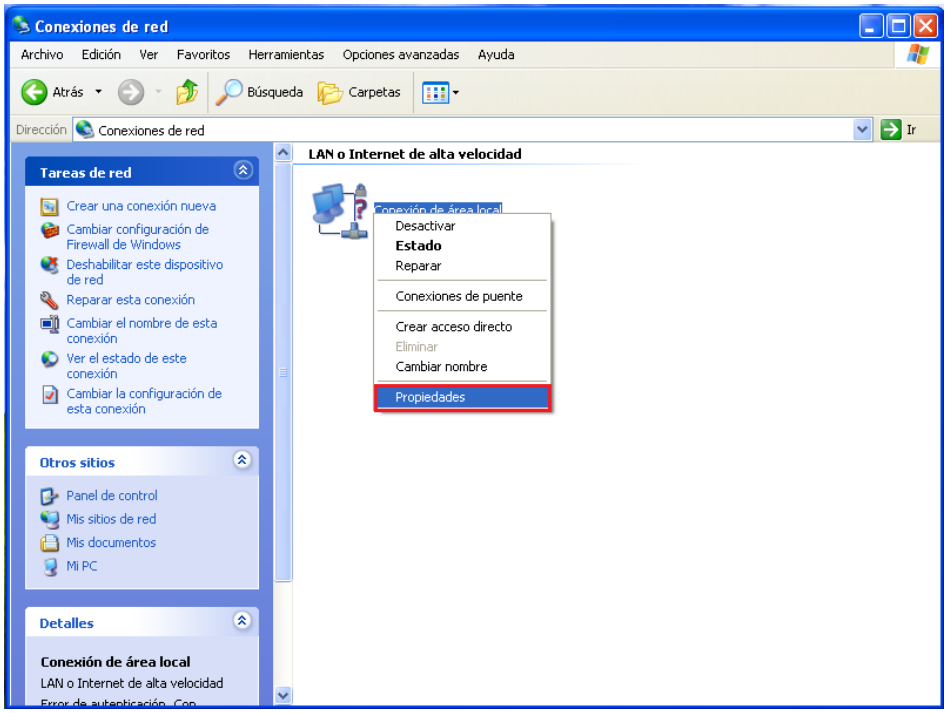


Després d'acceptar la confirmació d'instal·lació, el sistema operatiu ens indicarà que el certificat de la CA s'ha instal·lat en el sistema:

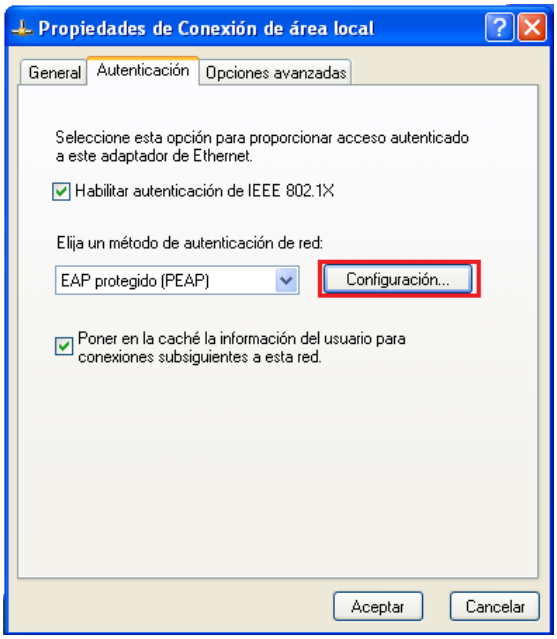


4.7.1.3 Configurar el Supplicant 802.1x de Microsoft Windows XP

"Inicio -> Panel de control -> Conexiones de red" i "*Conexión de área local* (botó secundari) -> *Propiedades":*

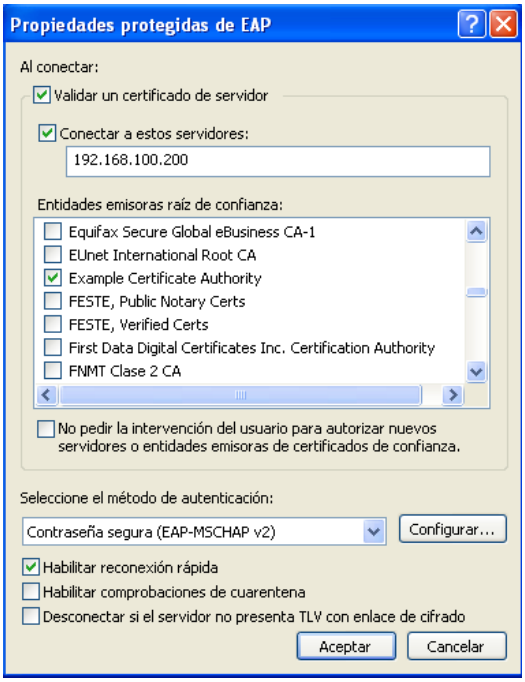


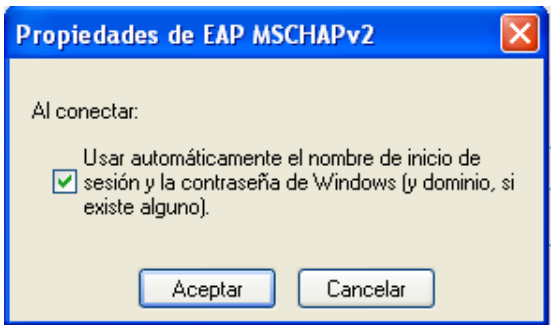
En la pestanya *"Autenticación"* seleccionem *"Habilitar autenticación de IEEE 802.1x"* i *"EAP protegido (PEAP)"* com a mètode d'autenticació. Seguidament seleccionem *"Propiedades"*.



Cal indicar al *Supplicant* que obtingui un certificat del servidor al realitzar l'autenticació. Per habilitar aquesta opció seleccionem *"Validar un certificado del servidor"*. En l'opció *"Conectar a estos servidores"* introduïm l'adreça IP del servidor. I en *"entidades emisoras de confianza"* escollim la CA afegida anteriorment.

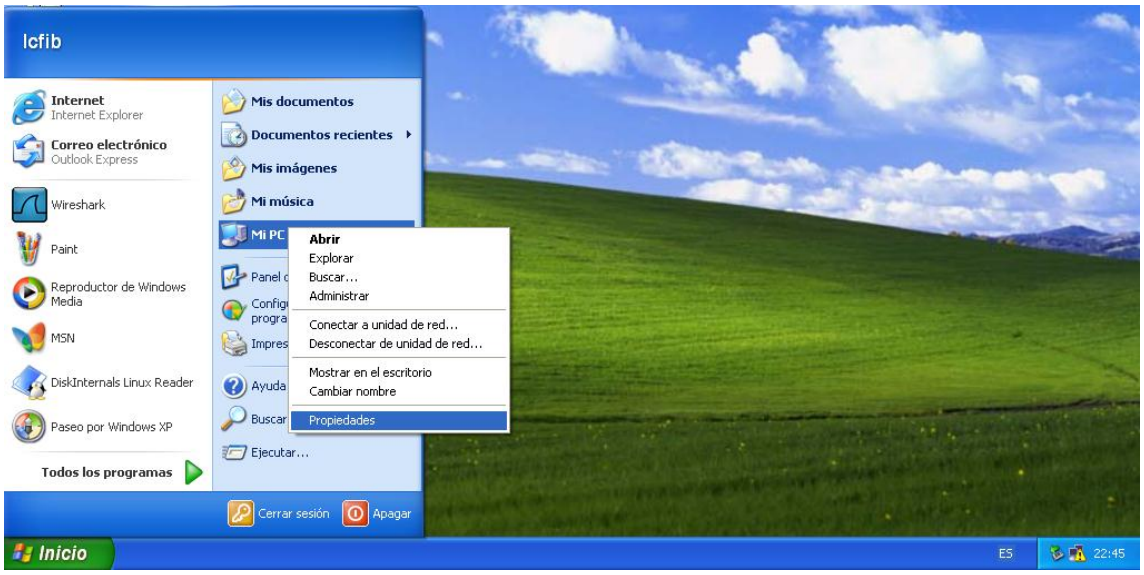
En l'opció *"Seleccione un método de autenticación"* escollim *"Contraseña segura (EAP-MSCHAPv2)"* i seleccionem *"Habilitar reconexión rápida"*. En el menú *"Configurar..."* habilitem l'opció *"Usar automáticamente el nombre de inicio de sesión y la contraseña de Windows (y dominio, si existe alguno)":*



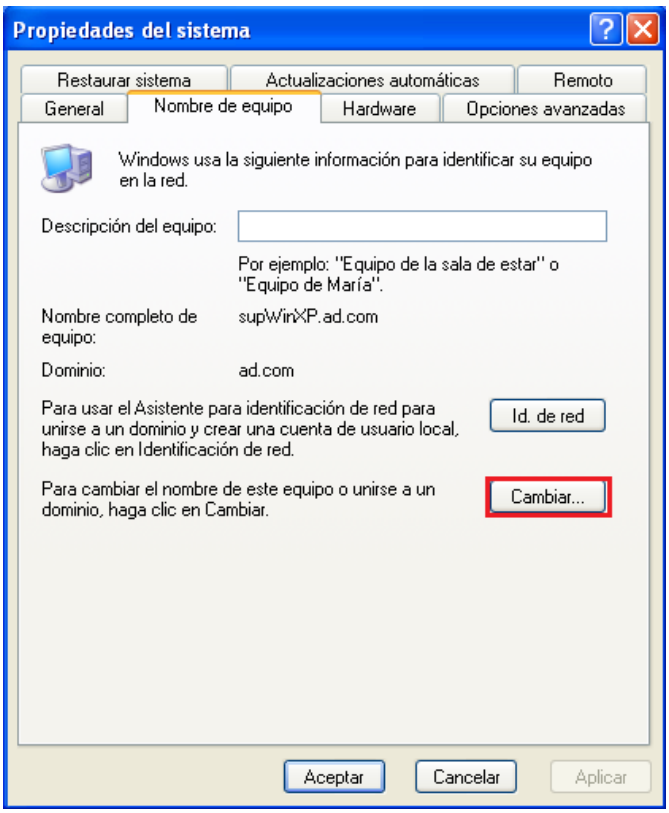


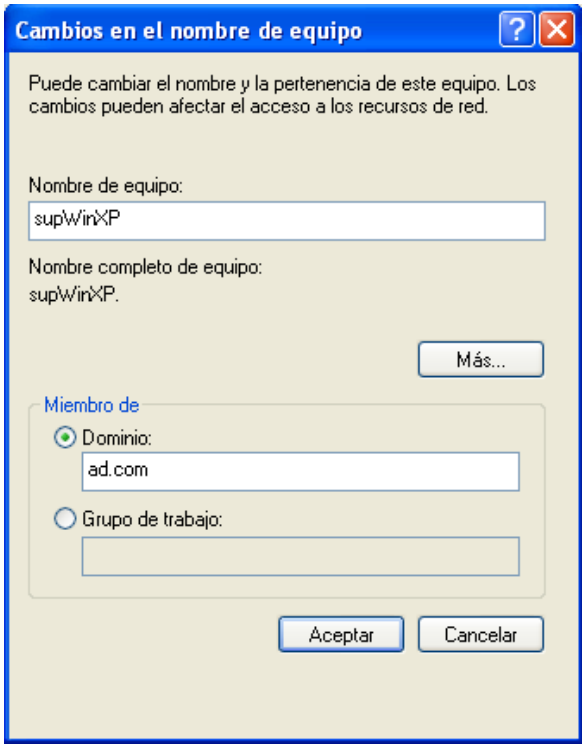
4.7.1.4 Afegir el Supplicant Windows XP a un domini Windows

"Inicio -> Mi PC (botó secundari) *-> Propiedades":*

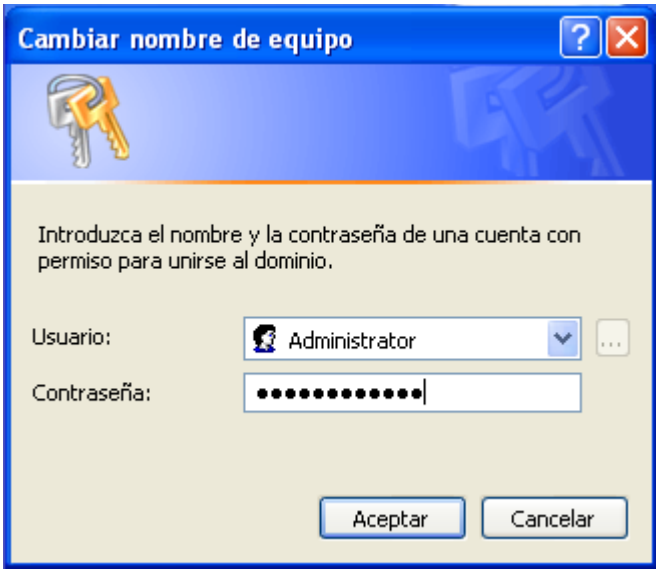


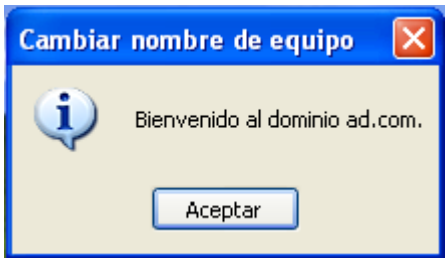
en la pestanya "*Nombre de equipo"* seleccionem el menú "*Cambiar..."* i en *"Dominio:"* introduïm el nom de domini *ad.com:*





En la pantalla emergent introduïm la contrasenya de l'administrador del domini:



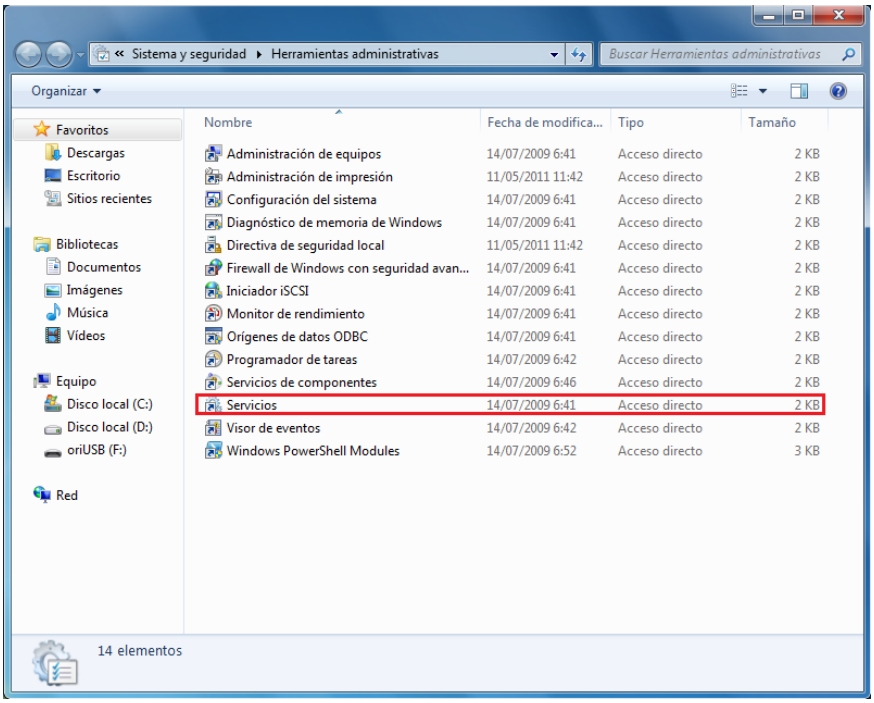


Després de reiniciar, el sistema ja formarà part del domini.

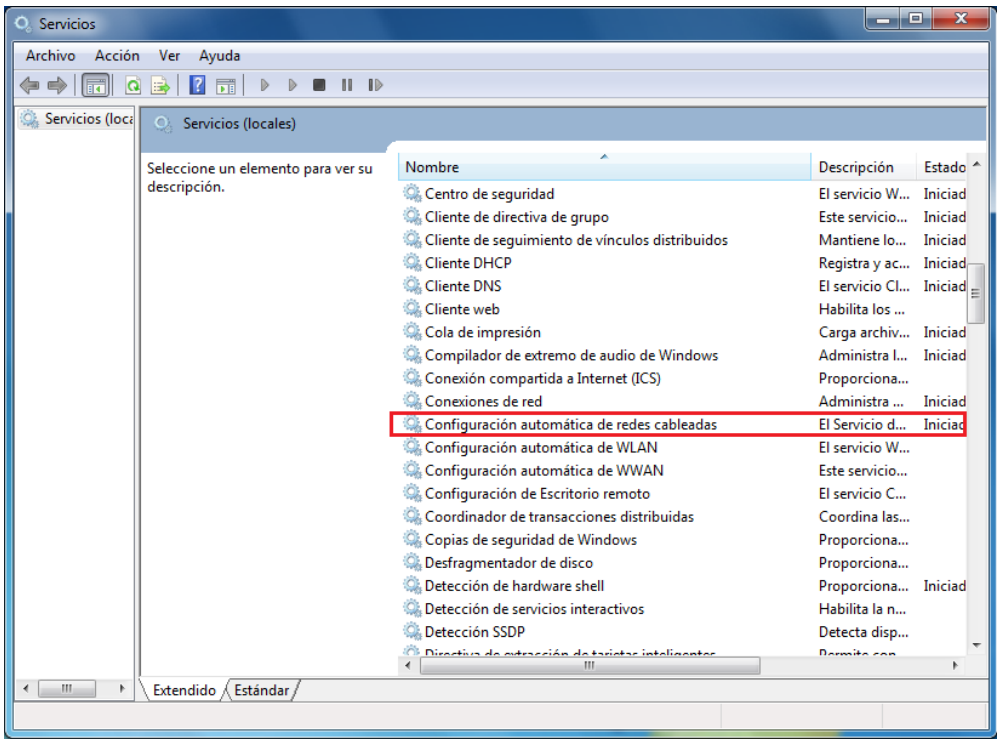
4.7.2 Windows 7

4.7.2.1 Habilitar l'autenticació 802.1x en Windows 7

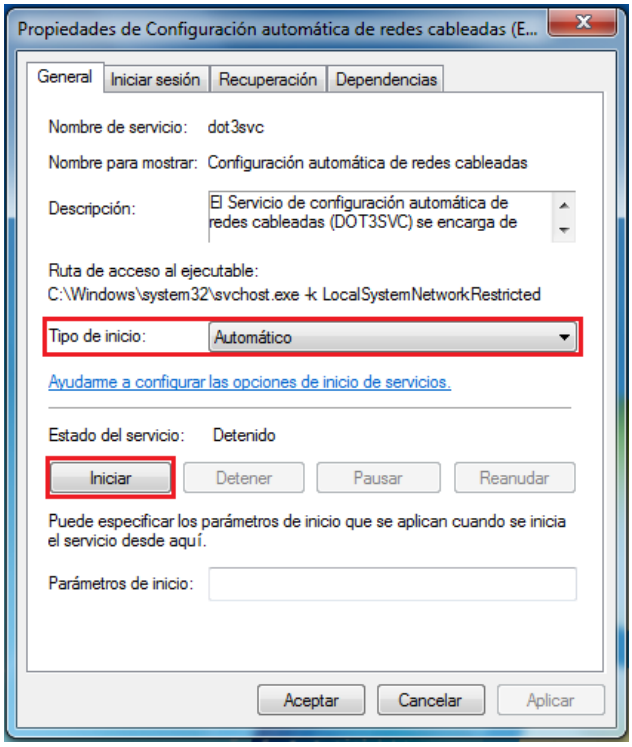
"Inicio -> Panel de Control -> Sistema y seguridad -> Herramientas administrativas -> Servicios":



Seleccionem "*Configuración automática de redes cableadas*":

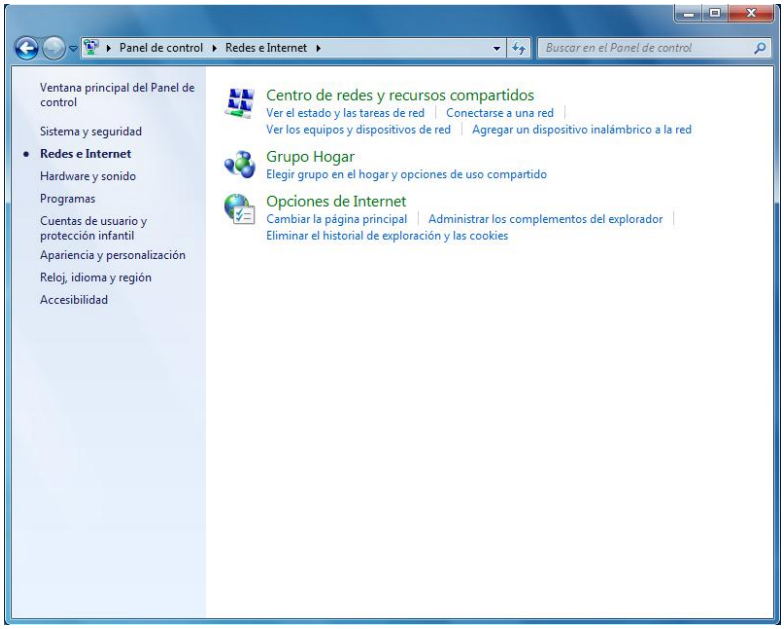


Escollim "*Tipo de Inicio : Automático*" i cliquem el botó "*Iniciar*":

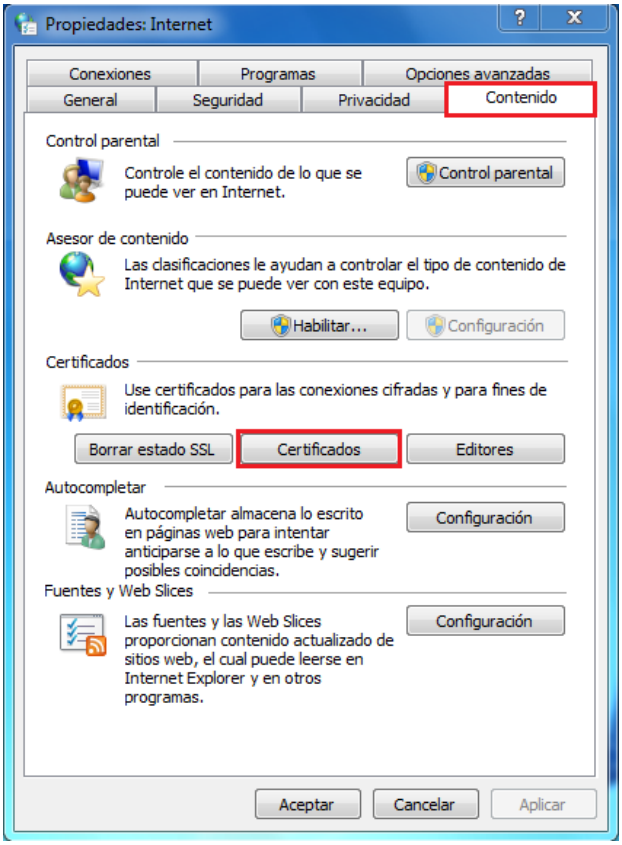


4.7.2.2 Afegir el certificat de l'Autoritat Certificadora en Windows 7

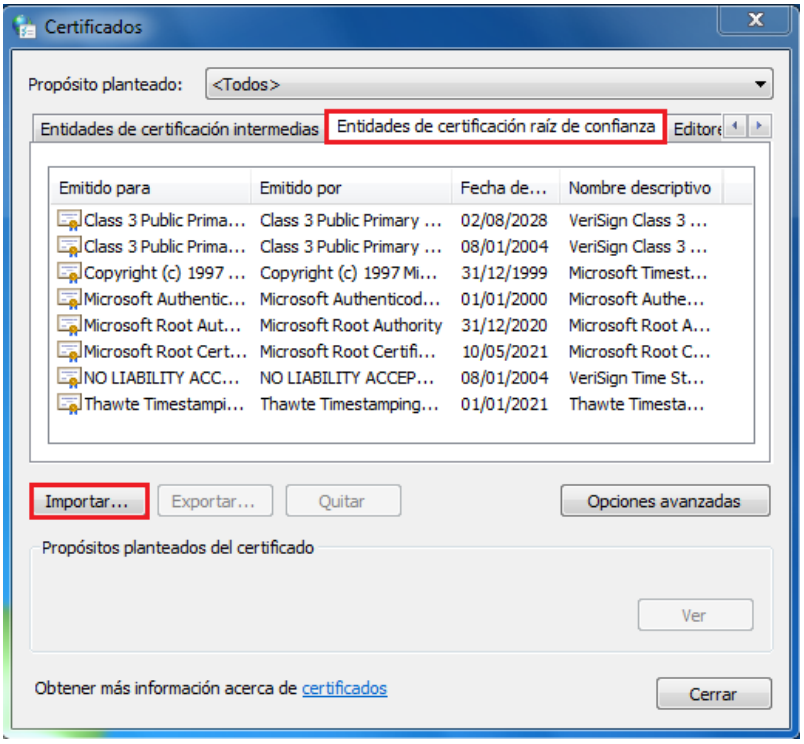
Ens dirigim al magatzem de certificats de Microsoft: "*Inicio -> Panel de control -> Redes e Internet -> Opciones de Internet*":



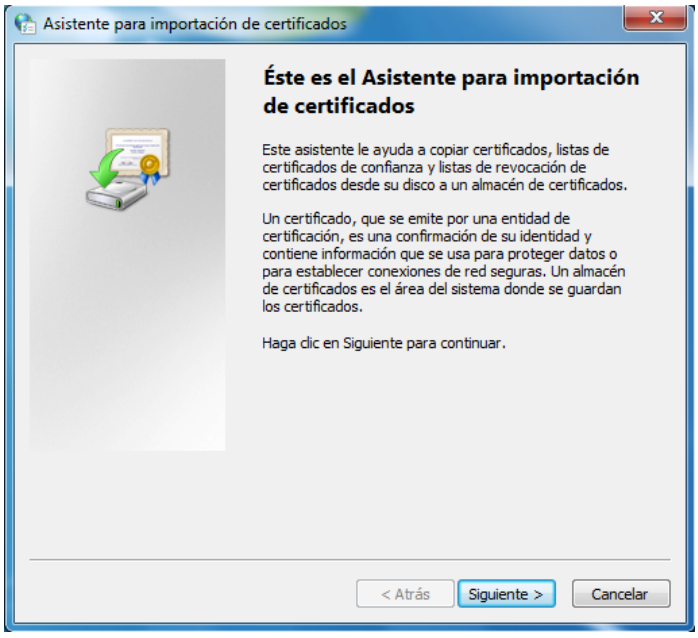
En la pantalla emergent seleccionem la pestanya "*Contenido*" i seleccionem "*Certificados*":

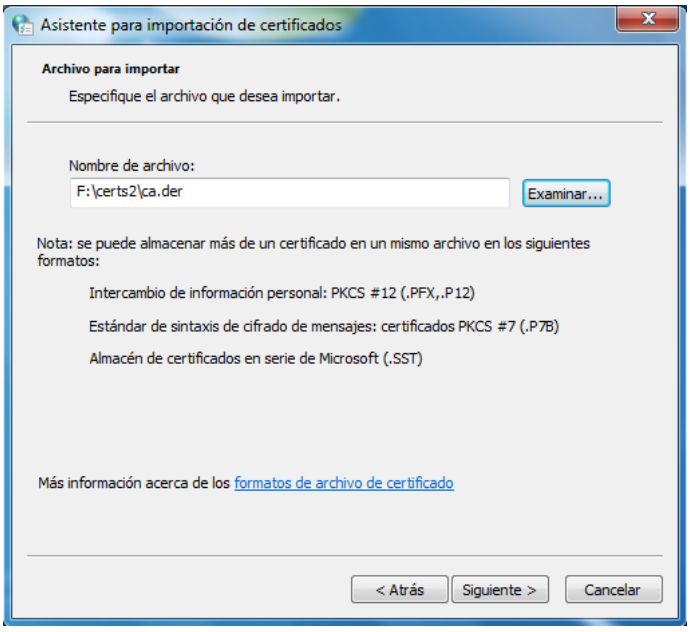


Un cop dins del magatzem de certificats, seleccionem "*Importar…*" en la pestanya "*Entidades de certificación raíz de confianza*":

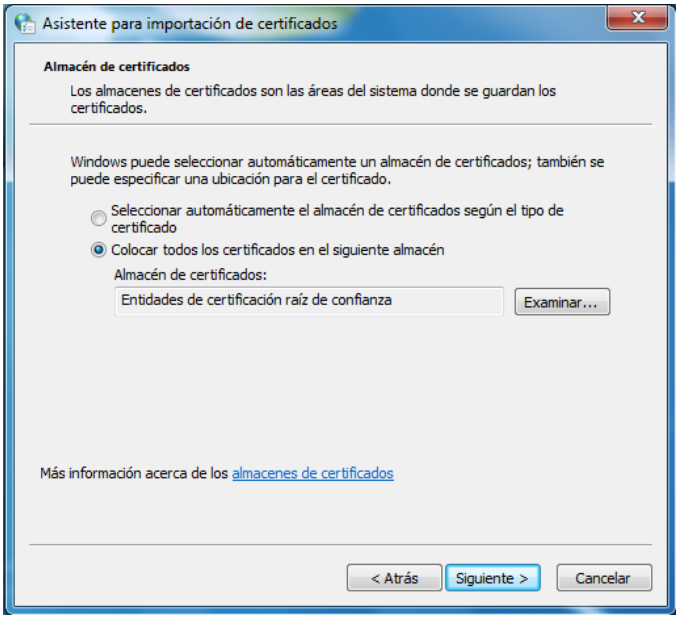


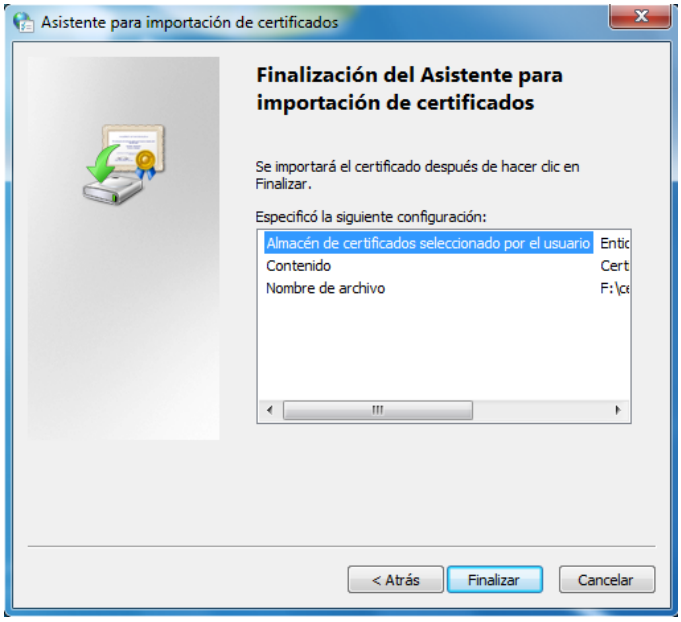
En la nova pantalla escollim "*Siguiente*" i indiquem la ruta on es troba el certificat arrel de la CA:



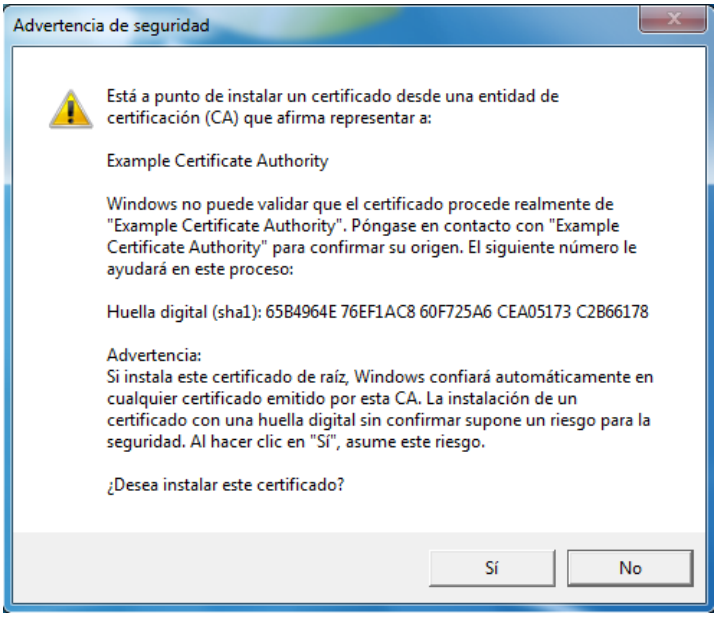


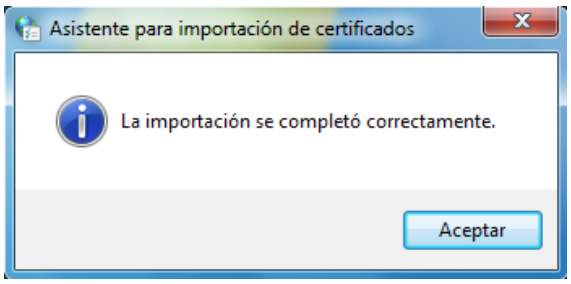
En l'opció "*Colocar los certificados en el siguiente almacén"* escollim "*Entidades de certificación raíz de confianza"* i finalitzem la instal·lació del certificat:





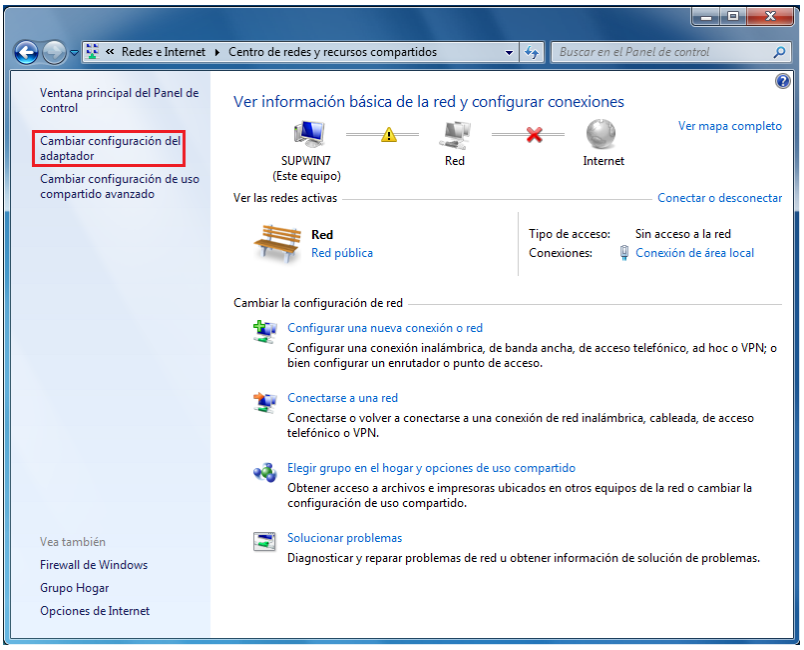
Un cop acceptada la confirmació requerida pel sistema, el certificat arrel de la CA ja estarà situat en el magatzem de certificats de Windows:



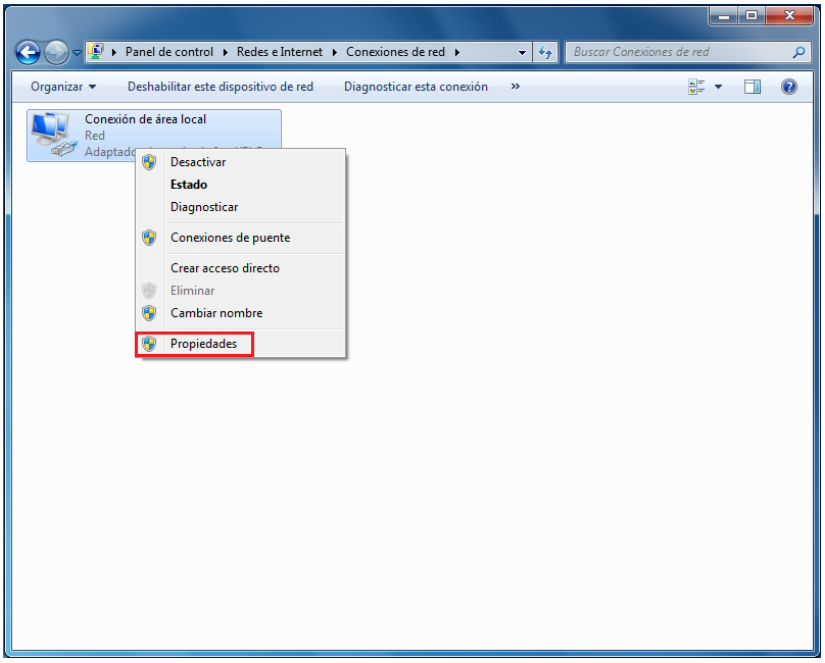


4.7.2.3 Configurar el Supplicant 802.1x de Microsoft Windows 7

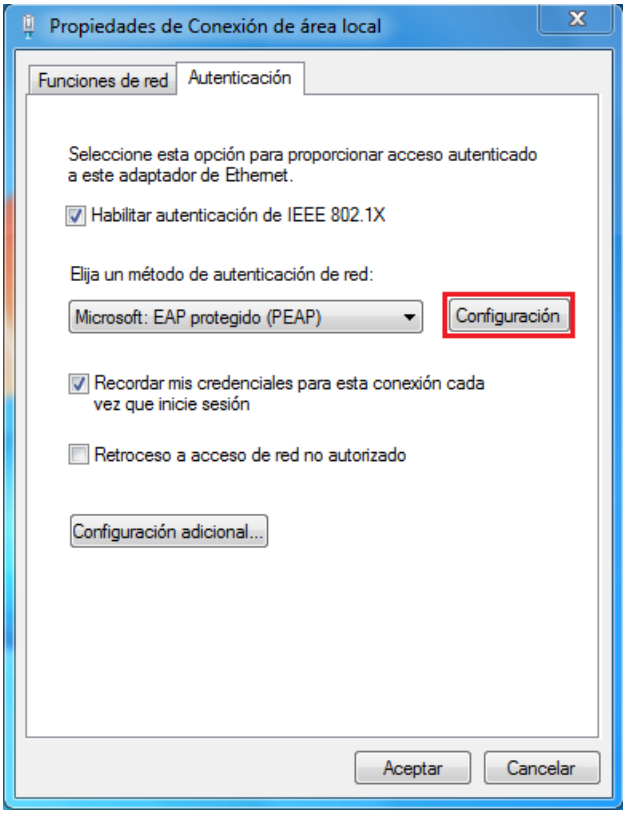
"Inicio –> Panel de control -> Centro de redes y recursos compartidos -> Cambiar la configuración del adaptador":



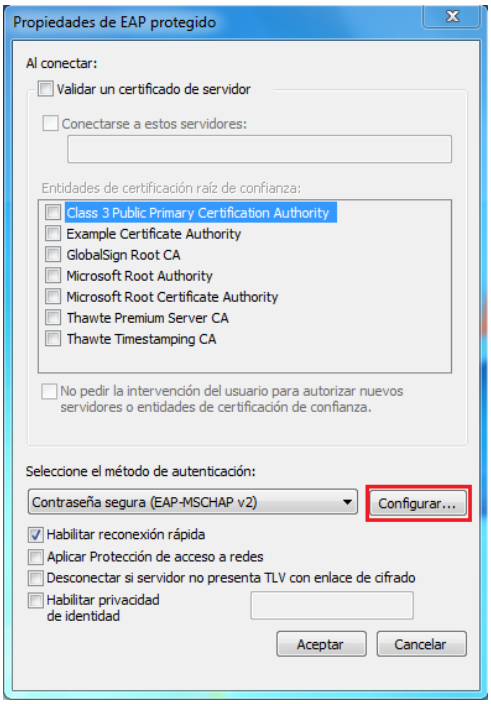
"Conexión de área local (botó secundari) *-> Propiedades":*

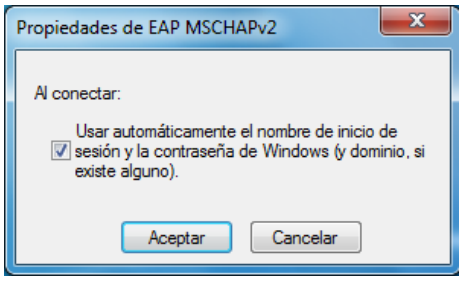


En la pestanya "*Autenticación*" seleccionem "*Habilitar autenticación IEEE802.1X*", "*Recordar mis credenciales para esta conexión cada vez que inicie sesión*" i "*Microsoft: EAP protegido (PEAP)*" com a mètode d'autenticació. Cliquem en el botó "*Configuración*":



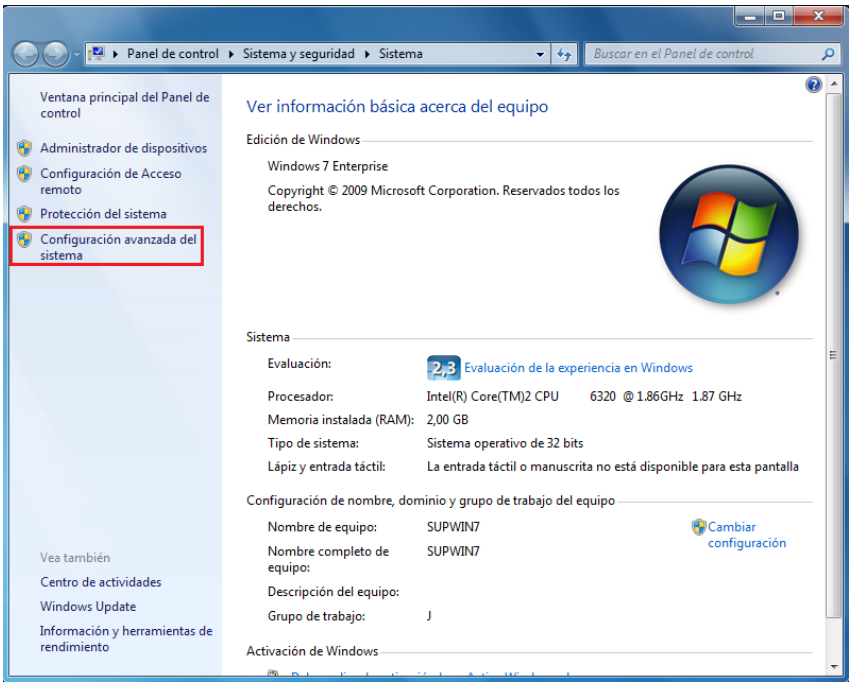
Escollim "*Contraseña segura (EAP-MSCHAP v2)*" com a mètode d'autenticació i "*Habilitar reconexión rápida*". Cliquem el botó "*Configurar…*" i en el menú emergent habilitem l'opció "*Usar automáticamente el nombre de inicio de sesión y la contraseña de Windows (y dominio, si existe alguno)*":



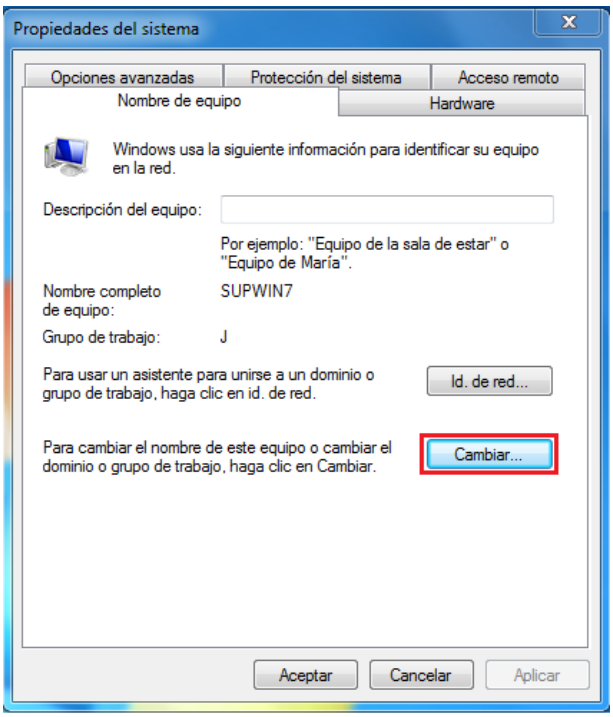


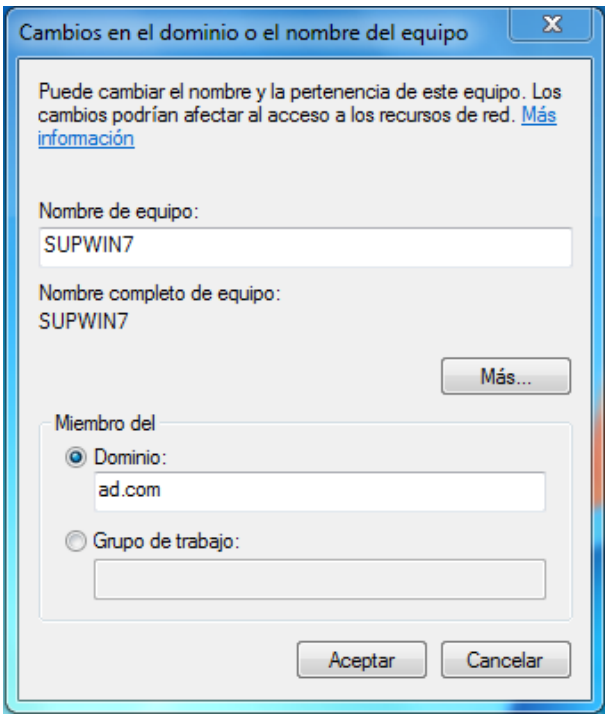
4.7.2.4 Afegir el Supplicant Windows 7 a un domini Windows

"Inicio -> Equipo (botó secundari) *-> Propiedades -> Configuración avanzada del sistema":*

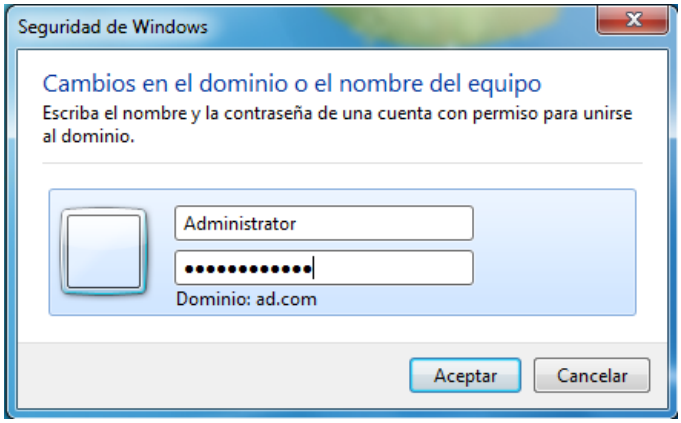


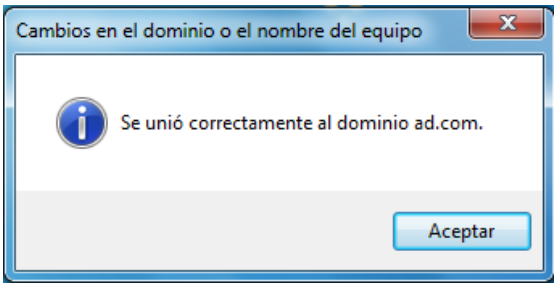
En la pestanya "*Nombre de equipo"* seleccionem "*Cambiar…"* i en la nova pantalla introduïm *ad.com* com a nom de domini:





En el menú emergent introduïm la contrasenya de l'administrador de domini:





4.7.3 Linux

4.7.3.1 Afegir el certificat de l'Autoritat Certificadora en Linux

Simplement cal col·locar el certificat arrel de la CA en el directori */etc/certs:*

```
$>cp ca.pem /etc/certs
```

4.7.3.2 Configurar l'autenticació 802.1x

Editem el fitxer de configuració de *l'wpa_Supplicant* */etc/wpa_Supplicant/wpa_Supplicant.conf* i li indiquem l'EAP-Method a utilitzar i la localització del certificat de la CA:

```
ctrl_interface=/var/run/wpa_supplicant

ctrl_interface_group=wheel


network={

key_mgmt=IEEE8021X

eap=PEAP

phase2="auth=MSCHAPV2"

ca_cert="/etc/certs/ca.pem"

anonymous_identity = "anonymous"
```

```
}
```

Si volem utilitzar el mètode d'autenticació EAP-TTLS enlloc de PEAP només cal substituir la línia *eap=PEAP* per *eap=TTLS*.

Creem l'script */usr/bin/network_auth.sh* i li donem permisos d'execució per a qualsevol usuari:

```
$>touch /usr/bin/network_auth.sh

$>chmod o+x /usr/bin/network_auth.sh
```

Editem l'script:

```
#!/bin/bash

#Reads password from PAM module
read PWD

#Kills all wpa_supplicant process (if any)
kill -9 `ps aux | grep wpa_supplicant | grep -v "grep wpa_supplicant" | awk '{print $2}'`
#Starts new wpa_supplicant process
/usr/sbin/wpa_supplicant -d -Dwired -i eth0 -c
/etc/wpa_supplicant/wpa_supplicant.conf &

#After 3 seconds, sends the username to wpa_supplicant
sleep 3
echo "IDENTITY 0 $PAM_USER" | /usr/sbin/wpa_cli &

# After 3 seconds, sends the user's password to wpa_supplicant
sleep 3
echo "PASSWORD 0 ${PWD}" | /usr/sbin/wpa_cli &

exit 0
```

Modifiquem el fitxer */etc/pam.d/common-auth* per afegir l'execució del mòdul PAM pam_exec.so:

```
#%PAM-1.0

#

# This file is autogenerated by pam-config. All changes

# will be overwritten.

#
```

```
# Authentication-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.). The default is to use the
# traditional Unix authentication mechanisms.
#
auth    required        pam_env.so
auth    required        pam_unix2.so
auth    required        pam_exec.so expose_authtok log=/var/log/network_auth.log
/usr/bin/network_auth.sh
```

El *flag expose_authtok* li indica al mòdul pam_exec.so que ha d'enviar la contrasenya com a paràmetre a l'script */usr/bin/network_auth.sh*

4.8 Proves

En aquest apartat es descriuen les proves que s'han realitzat en el pilot a mesura que s'hi afegien noves funcionalitats.

El comportament del sistema s'ha monitoritzat de varies formes:

- **Wireshark** S'ha instal·lat aquest *sniffer* de xarxa en el servidor i en els *Supplicants*. Ha ajudat a veure l'intercanvi de missatges 802.1x entre el *Supplicant* i l'*Authenticator*, i de trames RADIUS entre l'*Authenticator* i l'*Authentication Server* (servidor RADIUS).
- **freeRADIUS** El *daemon* del servidor RADIUS s'ha iniciat en mode *debug*. Aquesta opció permet veure els diferents fitxers i mòduls que el servidor va processant, les configuracions aplicades i les trames de resposta que envia a l'*Authenticator*.
- ***Authenticator*** Mitjançant una connexió Telnet a l'*Authenticator* s'ha pogut observar en tot moment les VLANs a les que estava associat cada port físic del commutador. D'aquesta forma es podia comprovar si les polítiques d'autorització escollides per l'*Authentication Server* eren aplicades per l'*Authenticator*.

A continuació es mostren les proves més representatives fetes en el sistema. Aquestes proves estan pensades i estructurades per servir de guia a tothom qui vulgui crear un sistema semblant a l'actual.

L'ordre en que es mostren les proves està basat en la implementació descrita al llarg del capítol 4 del present document. Les proves en el pilot s'han realitzat simultàniament amb l'etapa d'implementació, i es recomana seguir el mateix procediment en la implantació real del sistema. El sistema ha hagut d'executar satisfactòriament cadascuna de les proves suficients cops per poder afirmar que la supera.

Prova 1: Instal·lació de freeRADIUS i posada en marxa del daemon

Després d'instal·lar freeRADIUS tal i com s'explica en l'apartat 4.1 s'ha arrencat el *daemon* per primer cop. La dada més significativa a observar és la creació dels certificats de la CA, del servidor i del client. Aquest procés només s'executa de forma automàtica el primer cop que s'inicia el *daemon*.

Prova 2: La primera autenticació

Un cop comprovat que el *daemon* s'executa correctament s'ha realitzat la primera petició d'autenticació en el servidor. S'ha creat un usuari de proves en el fitxer *users* que s'autentica a través d'aquest mateix fitxer. Com a part de la fase d'autorització, freeRADIUS retorna a l'usuari un missatge de benvinguda si la fase d'autenticació finalitza satisfactòriament.

Com a client RADIUS s'ha utilitzat *radclient,* una eina proporcionada per freeRADIUS que permet realitzar proves d'autenticació de forma local sense necessitat de disposar d'un client RADIUS. En el fitxer de configuració de clients de freeRADIUS ja hi ha una entrada habilitada dedicada a acceptar les peticions locals, per tant no ha estat necessari fer modificacions en aquest fitxer.

Prova 3: Peticions d'autenticació fallides

Per a comprovar el comportament del servidor en cas d'error en la fase d'autenticació s'han realitzat peticions errònies per diferents motius: enviar un *shared secret* erroni, utilitzar una

contrasenya incorrecta i fer una petició d'autenticació per a un usuari inexistent. En tots tres casos, el servidor ha rebutjat la petició.

Prova 4: Peticions d'autenticació simple des d'un *Supplicant*

Aquesta prova s'ha realitzat utilitzant tota l'arquitectura del pilot. S'han instal·lat els sistemes operatius en el *Supplicant* i s'ha configurat el client RADIUS tal i com s'explica en l'apartat 4.2.

Abans de fer proves d'autenticació mútua (mitjançant certificats) s'han realitzat peticions utilitzant el mètode d'autenticació simple MD5.

Per defecte, freeRADIUS accepta MD5, per tant no ha estat necessari fer modificacions per realitzar aquesta prova.

L'usuari a autenticar ha estat el mateix que la prova anterior. Les peticions s'han enviat des dels tres sistemes operatius suportats (Windows XP, Windows 7 i Linux).

Prova 5: Peticions d'autenticació mútua des d'un *Supplicant* (PEAP, EAP-TTLS)

La següent prova ha consistit en realitzar peticions utilitzant els mètodes d'autenticació mútua (i únics suportats per al sistema) PEAP i EAP-TTLS. S'ha configurat el fitxer *eap.conf* del servidor i l'autenticació 802.1x dels *Supplicants* tal i com s'explica en els apartats 4.3 i 4.7 respectivament. En aquest punt, els *Supplicants* encara no s'han unit a cap domini.

Després de verificar que l'autenticació PEAP+MSCHAPv2 (Windows XP, Windows 7 i Linux) i EAP-TTLS+MSCHAPv2 (Linux) funciona correctament s'han realitzat noves peticions MD5 per tal de comprovar que freeRADIUS ja no accepta aquest mètode d'autenticació poc segur.

Prova 6: Autenticació en LDAP

El següent objectiu a assolir ha estat l'autenticació dels usuaris del pilot en LDAP. S'ha creat tota l'estructura LDAP descrita en l'apartat 4.4.1. Un cop afegits els tres usuaris (*oriol.bellet, prof, asac*) al servei de directori, s'han realitzat proves d'autenticació enviant peticions PEAP i EAP-TTLS des dels tres *Supplicants*.

Prova 7: Autenticació en Active Directory

Un cop testejada l'autenticació LDAP s'han realitzat les proves d'autenticació en Active Directory. La configuració del sistema per habilitar aquesta funcionalitat està descrita en l'apartat 4.4.2. Per tal de executar aquestes proves s'han afegit els *Supplicants* Windows XP i Windows 7 al domini Windows tal i com s'explica en els apartats 4.7.1.4 i 4.7.2.4 respectivament.

Prova 8: Verificar que freeRADIUS escull correctament el servei de directori un autenticar

Un cop el sistema és capaç d'autenticar usuaris tant en LDAP com en Active Directory, s'ha verificat que el sistema és capaç d'escollir correctament el servei de directori on autenticar l'usuari en funció de cada petició. FreeRADIUS realitza aquesta elecció en funció del domini al qual pertany el *Supplicant*. Per tant, per fer aquestes proves s'han anat afegint i treien els *Supplicants* del domini Windows.

Prova 9: Proves de reutilització de credencials

L'última prova significativa de la fase d'autenticació ha consistit en veure si els *Supplicants* són capaços d'iniciar el procés d'autenticació automàticament a l'iniciar sessió en el sistema operatiu sense necessitar la intervenció de l'usuari. La configuració d'aquesta funcionalitat ens els *Supplicants* està descrita en els apartats 4.7.1.3 (Windows XP), 4.7.2.3 (Windows 7) i 4.7.3.2 (Linux).

Prova 10: Autorització basada en el perfil de l'usuari

Per dur a terme aquesta prova s'ha configurat la fase d'autorització descrita en l'apartat 4.5, a excepció de l'apartat 4.5.1.3.1 on s'explica l'associació de les estacions de treball a un perfil. A l'haver passat per alt aquest pas, tots els *Supplicants* han estat tractats com a dispositius externs i s'ha pogut comprovar que els usuaris són assignats a la VLAN que els hi correspon en funció del seu perfil. Els usuaris *oriol.bellet, prof* i *asac* han estat col·locats en les VLANS STUDENTS_VLAN, PROFESSORS_VLAN i LCFIB_VLAN respectivament.

Prova 11: Autorització basada en el perfil de les estacions de treball

Un cop provat que l'autorització basada en el perfil de l'usuari s'executa correctament, s'ha realitzat l'apartat omès anteriorment (4.5.1.3.1), on s'han assignat perfils a les estacions de treball. D'aquesta forma es pot verificar l'autorització basada en estacions de treball.

Prova 12: Verificar que freeRADIUS aplica les autoritzacions segons el perfil de l'usuari o el de l'estació de treball de forma correcta

Un cop s'han realitzat suficients proves com per assegurar que el procés d'autoritzar en funció dels perfils funciona correctament, ha calgut comprovar que freeRADIUS aplica les polítiques d'autorització que pertoquen en cada cas. Les estacions de treball són autoritzades segons el seu perfil, mentre que els dispositius externs són autoritzats en funció del perfil de l'usuari que realitza la petició.

Prova 13: Autorització de les estacions de treball per arrencar per xarxa

Un cop les estacions de treball han estat assignades a perfils, tindran permisos per arrencar per xarxa. Aquest tipus d'autorització s'obté quan finalitza correctament el procés d'autenticació MAB. S'ha configurat la BIOS del *Supplicant* per indicar-li que el tipus d'arrancada és per xarxa. Mentre el *Supplicant* ha estat intentat arrencar per xarxa sense èxit, l'*Authenticator* ha realitzat l'autenticació MAB del dispositiu i la situat en una VLAN amb els recursos de xarxa suficients per arrencar.

Prova 14: Assignació de l'usuari a la GUEST_VLAN i a l'AUTH_FAIL_VLAN

L'última prova dedicada a l'autorització ha estat comprovar si els dispositius que no suporten (o no tenen habilitada) l'autenticació 802.1x són assignats a la GUEST_VLAN i si els usuaris que s'autentiquen erròniament 3 cops consecutius són situats a l'AUTH_FAIL_VLAN.

Per al primer cas, s'ha desactivat en els tres sistemes operatius l'opció d'enviar una petició d'autenticació quan s'inicia la sessió. Al no rebre resposta a les seves peticions d'autenticació, l'*Authenticator* ha situat els *Supplicants* en la GUEST_VLAN.

Per a provar la segona funcionalitat, s'han realitzat tres peticions d'autenticació invàlides. Després d'aquests tres intents, l'*Authenticator* ha col·locat el *Supplicant* en l'AUTH_FAIL_VLAN. Les peticions d'autenticació realitzades pel *Supplicant* un cop situat en aquesta VLAN han estat ignorades per l'*Authenticator*.

Prova 15: L'accounting

Per a configurar la fase *d'accounting* s'han seguit els passos descrits en l'apartat 4.6. Les proves han consistit en realitzar varies autenticacions de varis usuaris, en generar transit de xarxa i en comprovar si tota aquesta informació era emmagatzemada correctament en la base de dades.

Principals problemes trobats

En aquest apartat es resumeixen els principals problemes trobats i les solucions adoptades, la majoria dels quals ja s'han anat veient al llarg del document. La complexitat dels problemes és molt variada; alguns s'ha pogut solucionar en minuts, mentre que altres han requerit varis dies de dedicació.

Problema El servidor no accepta peticions del client tot i que les configuracions són correctes.

Solució El sistema operatiu *openSuSE* porta activat per defecte un tallafocs. Cal afegir noves regles per permetre al servidor escoltar peticions RADIUS del client.

Problema El servidor freeRADIUS pot autenticar usuaris en LDAP quan la petició es local, però no quan la realitza el *Supplicant*.

Solució El *Supplicant* es comunica amb el servidor mitjançant EAP-Methods. EAP només es compatible amb el xifrat de contrasenyes *NTLM_Auth*. Si s'utilitza qualsevol altre tipus de xifrat en LDAP, el procés d'autenticació fallarà.

Problema El servidor freeRADIUS és incapaç d'autenticar usuaris als quals se'ls hi ha eliminat el *hint* del seu *User-Name*.

Solució El nom d'usuari intervé en el xifrat EAP. Si es modifica, el procés d'autenticació donarà error. freeRADIUS permet utilitzar la variables especial *Stripped-User-Name* per emmagatzemar noms d'usuari als quals se'ls hi ha eliminat el *hint.*

Problema Cap *Supplicant* per a sistemes Linux és capaç de reutilitzar les credencials.

Solució wpa_supplicant disposa de l'eina *wpa_cli* que permet passar dades al binari wpa_supplicant en temps d'execució. Mitjançant el mòdul PAM *pam_exec* i utilitzant scripts, es pot capturar el nom d'usuari i contrasenya i passar-li al *Supplicant* a través de l'*wpa_cli.*

Problema Les estacions de treball necessiten arrencar per xarxa, però no es poden autenticar fins que el sistema operatiu ha arrancat.

Solució Es pot utilitzar l'autenticació MAB per permetre als dispositiu arrencar per xarxa. Un cop carregat el sistema operatiu, es pot tornar a autenticar l'usuari mitjançant 802.1x.

Problema Una estació de treball necessita dos tipus d'autorització diferents en funció de si necessita permisos per arrencar per xarxa o permisos convencionals. Això provoca que la taula de la base de dades dedicada a l'autorització tingui varies entrades (autoritzacions) per a cada dispositiu. Com pot saber freeRADIUS quina entrada (autorització) escollir en cada moment?

Solució Sense modificar la base de dades no pot. Hi ha dues solucions possibles al problema: Crear un nou camp en la taula indicant si el tipus d'autorització es convencional o per arrancar per xarxa (caldrà modificar les consultes SQL i les plantilles de creació de taules), o crear dos taules (una dedicada a l'autorització convencional i l'altra a l'autorització per arrencar per xarxa). En el sistema implementat s'ha escollit la segona.

Problema Com pot saber freeRADIUS si una estació de treball vol una autorització per arrencar per xarxa o una autorització convencional?

Solució Les peticions d'autorització per a poder arrencar per xarxa venen en trames MAB (no EAP). Les peticions d'autorització convencionals venen en trames 802.1x (EAP). Quan freeRADIUS rebi una petició, haurà de comprovar si el missatge ve encapsulat en una trama EAP.

Problema Com sap freeRADIUS si ha d'autenticar un usuari en un servei de directori LDAP o en Active Directory?

Solució Els usuaris que s'han d'autenticar en Active Directory pertanyen al domini *AD***.** Quan un *Supplicant* pertany a un domini, envia el nom del domini com a prefix en l'atribut *User-Name***.** FreeRADIUS sabrà el servei de directori on autenticar comprovant si el camp User-Name comença per *AD/*.

5. Conclusions

En aquest capítol es descriuen les conclusions resultants de la realització del projecte. En primer lloc, es mostra una comparativa entre el temps real dedicat a la realització de cada etapa i la planificació inicial. També es dedica un apartat a l'anàlisi econòmic, on es calcula el cost que ha tingut la realització del projecte. Finalment, a més de comprovar l'assoliment d'objectius i fer una valoració personal del projecte, s'analitza l'impacte i els beneficis que tindria implantar el sistema de control d'accés en un entorn real.

5.1 Dedicació al projecte

En aquest apartat es mostra un comparativa entre la planificació inicial del projecte i el temps final dedicat a la realització de cada tasca.

En la figura 5.1 es mostra un nou diagrama de Gantt on s'il·lustra la repartició de la dedicació entre les diferents etapes.

Tal i com s'ha fet en el capítol inicial d'aquest document, també s'ha construït una taula on es desglossa l'esforç invertit en cada etapa. La taula 5.2 serà útil per fer una comparativa entre la planificació inicial i la dedicació real. Les dades aquí mostrades poden no ser completament exactes, però són estimacions molt acurades.

En aquesta taula també s'ha dividit l'etapa de Documentació en dues parts degut a que el percentatge de dedicació a aquesta tasca ha variat al llarg del projecte.

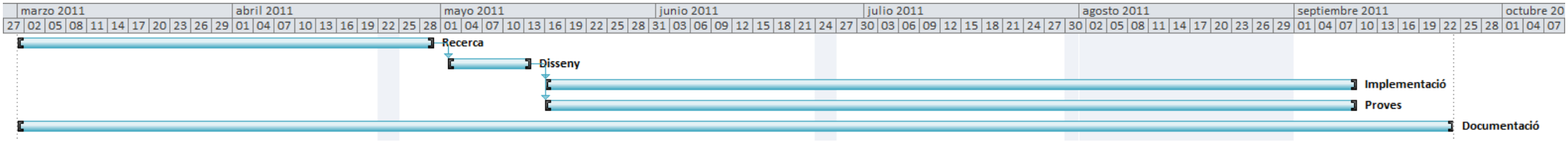


Figura 5.1 – Dedicació al projecte

Etapa	Data Inici	Data Fi	Duració (dies)	Dedicació	Duració efectiva (dies)	Duració efectiva (hores)	% temps dedicat
Recerca	01/03/2011	01/05/2011	43	90%	38.7	309.6	31.464%
Disseny	02/05/2011	15/05/2011	10	90%	9	72	7.317%
Implementació	16/05/2011	11/09/2011	61	65%	39.65	317.2	32.236%
Proves	16/05/2011	11/09/2011	61	25%	15.25	122	12.398%
Documentació (1)	01/03/2011	11/09/2011	114	10%	11.4	91.2	9.268%
Documentació (2)	01/09/2011	23/09/2011	9	100%	9	72	7.317%

Taula 5.2 – Temps dedicat a la realització de cada etapa

5.1.1 Desviacions sobre la planificació inicial

En planificació inicial del projecte es contemplava una duració total de 128 dies (1024 hores). Finalment, el projecte s'ha pogut concloure una setmana abans del previst. La dedicació ha estat de 123 dies treballats, unes 984 hores.

Durant la realització del projecte, s'ha tingut present i s'ha intentat respectar la planificació inicial en tot moment. Una mostra d'això és que el temps invertit en les dues primeres etapes (Recerca i Documentació) ha coincidit plenament amb les previsions fetes.

Les primeres desviacions respecte a la planificació inicial es van començar a notar durant l'etapa d'implementació. Sempre és difícil fer una estimació temporal d'aquesta etapa, ja que la seva realització depèn en bona part del nombre de problemes inesperats que puguin sorgir i el temps que costi trobar-ne una solució. S'han invertit 62,4 hores més de les previstes per a la realització d'aquesta etapa.

L'etapa de proves també ha tingut una durada molt similar a la prevista, només hi ha hagut una desviació de 12 hores. Les proves s'han realitzat simultàniament amb la implementació.

5.2 Anàlisi econòmic

En aquest capítol es descriu el cost econòmic que ha tingut el projecte realitzat. El cost es divideix en tres parts:

- Cost de personal
- Cost software
- Cost hardware

En els següents apartats es detallarà cadascun d'aquests costs i el total.

5.2.1 Cost de personal

En aquest apartat es detallen els costos dels recursos personals dedicats al projecte. Les tasques realitzades s'han dividit en les mateixes quatre en les que s'ha dividit el projecte: Recerca, Disseny, Implementació i Proves . S'ha afegit també el cost resultant de redactar aquest document. El cost preu/hora (7 €/h) sorgeix dels ingressos nets d'un becari en l'LCFIB.

Tasca	Preu / Hora	Hores	Cost
Recerca	7 €/h	309.6	2.167,20 €
Disseny	7 €/h	72	504,00 €
Implementació	7 €/h	317.2	2.220,40 €
Proves	7 €/h	122	854,00 €
Documentació	7 €/h	163.2	1.142,40 €
		Cost de personal total	**6.888,00 €**

5.2.2 Cost hardware

El hardware utilitzat per a la realització del projecte ha estat facilitat per l'LCFIB i no ha estat necessària la seva adquisició. El cost s'ha calculat en base al preu del recurs en el moment de la compra i la amortització que se n'ha fet durant la realització del projecte. En la següent taula es desglossa el resultat obtingut:

Recurs Hardware	Quantitat	Preu / Unitat	Cost	Anys amortització	Cost / Mes	Durada projecte	Cost amortitzat
HP Compaq 8100	4	752,22 €	3008,88 €	3	83,58 €	6 mesos	501,48 €
Monitors CRT 17"	4	78,06 €	312,24 €	3	8,67 €	6 mesos	52,04 €
Cisco Catalyst 3550	1	2.296,00 €	2.296,00 €	7	27,33 €	6 mesos	164,00 €
						Cost hardware total	**717,52 €**

5.2.3 Cost software

En la següent taula es mostra el cost dels diferents components software que s'han utilitzat per realitzar el projecte.

Recurs software	Tipus llicència	Cost
Open SuSE	GNU GPL	0 €
Windows Server 2008	Llicència UPC	0 €
freeRADIUS	GNU GPL	0 €
Windows XP	Llicència UPC	0 €
Windows 7	Llicència UPC	0 €
MySQL	GNU GPL	0 €
openLDAP	OpenLDAP Public License	0 €
Active Directory	Llicència UPC	0 €
Microsoft Office 2010	Llicència UPC	0 €
Krb5	GNU GPL	0 €
Samba	GNU GPL	0 €
Wpa_supplicant	GNU GPL	0 €
Microsoft Project	Llicència UPC	0 €
Wireshark	GNU GPL	0 €
	Cost software total	**0 €**

Com s'observa, el cost software és 0€. Això es degut a que tot software i sistema utilitzat té llicència GNU GPL (General Public License) o llicència UPC, que permet al personal de la universitat obtenir llicències gratuïtes de software de pagament. Realment, la Llicència UPC sí suposa un cost econòmic per a la universitat, però aquesta quantitat és tant baixa i difícil de calcular amb exactitud (les llicencies es compren per paquets i van subjectes a altres condicions), que s'ha preferit ignorar-la.

Per altra banda, s'han pogut obtenir llicències de productes Microsoft gràcies a l'acord MSDNAA subscrit entre la Facultat i aquesta companyia, que permet als alumnes de la FIB obtenir gratuïtament diferent software de Microsoft.

5.2.4 Cost total

Per a calcular el cost total del projecte simplement s'han sumat els tres costos calculats anteriorment:

	Cost
Cost de personal	6,888,00 €
Cost software	0,00 €
Cost hardware	717,52 €
Cost total del projecte	**7.605,52 €**

5.3 Assoliment dels objectius

En aquest apartat es recuperen els objectius inicials del projecte i es mostra la forma en que s'han assolit:

- **Autenticació d'usuaris** La validació de la identitat dels usuaris es realitza en varis serveis de directori. La font on l'usuari ha de ser autenticat depèn del perfil del dispositiu des del qual l'usuari vol accedir a la xarxa. Si l'usuari està connectat en una estació de treball del Laboratori de Càlcul de la Facultat, s'autenticarà en un Active Directory. En la resta de casos (estacions de treball de les aules i dispositius externs) els usuaris s'autenticaran en un LDAP. Quan el sistema rep una petició d'autenticació és capaç de deduir el perfil del dispositiu que demana accés i, per tant, d'autenticar l'usuari en la font de dades adient.
- **Definir els recursos que s'oferiran a cada usuari** S'han creat varis perfils d'usuaris i dispositius. Els recursos de xarxa que se li oferiran a cada usuari seran en funció del seu perfil o del perfil del dispositiu al qual està connectat.

  Per exemple, quan un estudiant estigui realitzant tasques des d'un ordinador d'un aula del laboratori tindrà els permisos que li corresponen com a alumne que està realitzant una activitat docent en aquell precís instant. En canvi, quan connecti el seu propi ordinador portàtil a qualsevol punt d'accés se li oferiran els recursos de xarxa globals que li corresponen com a estudiant de la facultat. Professors i membres del laboratori de càlcul també tindran un perfil, que serà diferent que el dels estudiants. Als individus que no siguin membres de la facultat també se'ls hi permetrà accedir a la xarxa mitjançant els seus propis dispositius, però els recursos que se'ls hi oferiran seran més reduïts, ja que seran tractats com a convidats.
- **Mantenir un registre d'activitat de cada usuari** El sistema es capaç d'emmagatzemar dades sobre l'ús que fan els usuaris dels recursos de xarxa que se li han ofert. Això permetrà, per exemple, crear gràfiques i estadístiques que ajudaran a observar la forma en la que la xarxa és utilitzada, ja sigui col·lectiva o individualment.
- **Monitoritzar el comportament del sistema** Actualment el Laboratori de Càlcul compta amb un entorn de centralització i anàlisi de *logs*. El sistema de control d'accés a la xarxa podrà ser afegit a aquest entorn. D'aquesta forma, els *logs* generats pels commutadors i servidors del sistema seran enviats a un servidor on s'analitzaran automàticament. El resultat d'aquest anàlisi serà traslladat a un sistema de control Nagios que podrà alertar de possibles caigudes en el servidor, de funcionaments anòmals i de possibles intents d'atac.

5.4 Impacte resultant de la implantació del sistema de control d'accés

Un cop finalitzat el projecte i havent verificat que el sistema pilot compleix tots els objectius pels quals es va desenvolupar, es pot realitzar una valoració sobre l'impacte que suposaria la implantació del sistema de control d'accés a la xarxa cablejada.

Durant la realització del projecte, les decisions de disseny presses han estat sempre en funció de com està actualment organitzada la xarxa cablejada de la facultat. Es va tenir en compte aquesta consideració precisament per evitar que l'impacte que tindria la implantació del sistema de control d'accés fos elevat.

Gràcies al pilot desenvolupat, s'ha pogut constatar que l'impacte que tindria la posada en marxa del sistema de control d'accés no seria significatiu. No caldria afegir nous dispositius de xarxa ni servidors, no caldria adquirir nou software, no suposaria cap molèstia pels usuaris, etc.

L'únic cost que tindria seria a nivell de personal. La implementació d'un control d'accés no és trivial, i tot que en aquest document es dedica un capítol sencer a explicar com s'hauria de dur a terme aquesta implementació, el nombre de tasques a realitzar és elevat, així com la seva complexitat.

En la següent taula es mostren les principals tasques que s'haurien de realitzar per a poder implantar el sistema en l'entorn real de la facultat. Per a cada tasca, s'ha realitzat una estimació del cost temporal que podrien tenir:

Tasca	Hores
Instal·lació del software (MySQL, OpenLDAP, OpenSSL, FreeRADIUS, securització i proves)	20
Configuració dels tallafocs	5
Configuració dels commutadors	
Integració dels commutadors en una arquitectura AAA	25
Configuració en els commutadors	5
Configuració en freeRADIUS	5
Configuració del mòdul EAP en freeRADIUS i creació de certificats	20
Fase d'autenticació	
Configurar el mòdul LDAP i l'autenticació en freeRADIUS	3
Configurar OpenLDAP, Samba, krb5 i Active Directory	15
Integrar els servidors en els dominis de Windows	1
Fase d'autorització	
Configurar el mòduls SQL i l'autorització en freeRADIUS	40
Crear les bases de dades MySQL i les taules	2
Integració de les bases de dades amb els sistemes d'informació actuals	40
Fase d'*accounting*	
Configurar l'*accounting* en freeRADIUS	10
Crear les taules d'*accounting* en MySQL	2
Estudi del funcionament en alta disponibilitat	30
Instal·lació del servidor de reserva	50
Estudi de la monitorització del sistema	30
Integració amb els sistemes actuals	10
Redacció de la documentació sobre el sistema de control d'accés	20
Formació del personal que gestionarà el sistema de control d'accés	30

Total	**182**

Com s'observa, la implantació del sistema de control d'accés en tota la facultat es podria realitzar en un termini de 182 hores, que equivalen aproximadament a un mes de feina.

Per últim, també s'han analitzat la forma en la que s'hauria de realitzar la implantació del sistema de control d'accés en l'entorn real. S'han dissenyat principalment dues fases:

- **Fase 1** Consistiria en la implantació del sistema en les principals sales de l'edifici B6: la Sala d'actes i la Sala de juntes.

  En aquestes sales solen accedir a la xarxa tan estacions de treball què estan allí situades de forma permanent, com dispositius externs que porten els propis usuaris. El perfil d'aquest usuaris també es variat.

  La heterogeneïtat del tipus d'usuaris i dispositius que accediran a la xarxa en aquesta fase permetrà verificar totes les funcionalitats del sistema treballant conjuntament.
- **Fase 2** Un cop verificat el comportament del sistema en aquesta primera fase, ja es podria realitzar la segona (i darrera), en la que s'implantaria el control d'accés a la resta d'edificis de la FIB.

5.5 Beneficis resultants de la implantació del sistema de control d'accés

El projecte realitzat ha permès demostrar que la implantació d'un sistema de control d'accés a la xarxa de la Facultat no només és viable, sinó que aportaria grans beneficis tant a nivell de flexibilitat com de seguretat.

Tots els membres de la Facultat, i també aquells individus que no hi formen part, podrien accedir a la xarxa cablejada des de qualsevol punt d'accés de la facultat, conservant en tot moment els privilegis propis del seu perfil.

També les estacions de treball es beneficiarien de la flexibilitat del sistema. Per exemple, quan un ordinador s'espatlli i hagi de ser substituït per un altre, o simplement es canviï la seva targeta de xarxa, no serà necessari realitzar cap tipus d'acció per permetre l'accés al nou dispositiu. El sistema el reconeixerà automàticament i li oferirà els recursos adients.

Un altre exemple molt aclaridor és el que es produeix en les èpoques de matriculació dels estudiants. Per realitzar el procés de matrícula s'utilitzen impressores, que són connectades a la xarxa a través dels punts d'accés situats a les aules del laboratori, on es realitza la matriculació. En períodes fora de matricula, aquells punts d'accés són assignats a unes estacions de treball concretes del laboratori. Quan arriba la matriculació i cal connectar les impressores a aquests punts d'accés, un membre del Laboratori de Càlcul ha de modificar manualment les configuracions dels commutadors afectats, assignant els seus ports físics a les impressores, i col·locant-les en les VLANs apropiades. Aquesta tasca pot arribar a tenir un cost d'una hora.

Amb el sistema de control d'accés a la xarxa dissenyat, tot aquest procés es realitzaria de forma automàtica. El sistema sabria de l'existència de les impressores, i de l'VLAN a la que s'haurien de col·locar. Per tant, quan una impressora fos connectada a qualsevol punt d'accés de la facultat, el sistema li donaria accés a la xarxa i la col·locaria a l'VLAN que li pertoqués de forma automàtica.

Actualment, no es pot saber quin usuari està fent ús de la xarxa en cada moment. Amb la implantació del nou sistema, tot usuari que volgués accedir a la xarxa necessitaria identificar-se mitjançant un nom d'usuari i una contrasenya. D'aquesta forma, es podrien restringir el privilegis de xarxa en funció de l'usuari que hi estigués accedint, no només del dispositiu. Però aquest augment de seguretat no suposaria un augment de complexitat per a l'usuari. Quan l'usuari es volgués connectar a la xarxa a través d'un ordinador de les aules o de l'LCFIB, el procés d'autenticació es duria a terme d'una forma transparent, ja que el sistema operatiu utilitzaria les credencials introduïdes per l'usuari a l'iniciar sessió per autenticar-lo a la xarxa, sense que fos necessària la seva intervenció.

El sistema també seria capaç d'emmagatzemar dades estadístiques sobre l'ús que fa cada usuari de la xarxa. Aquestes dades podran servir per fer diferents gràfiques d'ús, com per exemple saber el trànsit que es genera al llarg d'un dia, o quines hores són les de més ocupació, o quins punts d'accés solen utilitzar els usuaris per connectar els seus portàtils, etc.

Un dels aspectes més importants del sistema desenvolupat és el fet que està basat en estàndards d'ús molt comú en tot tipus de xarxes. Això permet que la implantació d'aquesta solució en un altre tipus de xarxa, com per exemple les sense fils, sigui pràcticament automàtica.

Per últim, cal remarcar que el sistema aquí presentat no té un ús únic i exclusiu de la facultat. Amb una senzilla adaptació es podria utilitzar com a control d'accés en un gran nombre d'organitzacions, com per exemple empreses (on cada cop és més comú) i escoles, tant per xarxes cablejades com per xarxes sense fils.

5.6 Conclusions personals

Per posar el punt i final exposaré l'aportació a nivell personal que ha suposat la realització del projecte i del Màster en Tecnologies de la Informació.

Sota el meu punt de vista, la principal aportació del projecte ha estat la experiència adquirida en vàries tecnologies i sistemes sobre les quals no havia tingut l'oportunitat de treballar prèviament. Alguns exemples són LDAP, Windows Server o Active Directory.

Un altre aspecte enriquidor ha estat l'etapa de recerca, que m'ha permès descobrir quins mecanismes d'autenticació a la xarxa existeixen, i quines possibilitats permeten. Abans de la realització del projecte, els coneixements en aquests camps eren generals i ambigus.

Els coneixements adquirits durant el Màster m'han ajudat de maneres diverses en la realització del projecte. Les assignatures que considero que m'han enriquit més són aquelles enfocades als sistemes operatius i a les xarxes, com poden ser EDSO (Estructura i Disseny de Sistemes Operatius), SODX (Sistemes Operatius Distribuïts en Xarxa), PIAM (Protocols d'Internet i Aplicacions Multimèdia) o SPD (Serveis Públics de Dades).

Altres no m'han ajudat directament en el projecte però m'han ofert una base que pot ser molt valuosa en un futur professional. La introducció en àmbits com els compiladors, la intel·ligència artificial o l'arquitectura de computadors (per posar tres exemples) m'ha ajudat a tenir una visió més àmplia sobre les tecnologies de la informació i la informàtica en general.

Per últim, i no per això menys important, cal remarcar assignatures pràctiques com PROSO (PROjecte de Sistemes Operatius) i PROP (PROjecte de Programació), que m'han permès adquirir més experiència en el camp de la programació.

Bibliografia

[1] Brown, Edwin Lyle. *802.1x Port Based Authentication*. Auerbach Publications Taylor & Francis Group. 2007. ISBN(13): 978-1-4200-4464-5. ISBN(10): 1-4200-4464-8.

[2] Fernández Hansen, Yago. Ramos Varón, Antonio. García-Morán, Jean Paul. *RADIUS / AAA / 802.1x: Sistemas basados en la autenticación en Windows y GNU/Linux. Seguridad Máxima.* RA-MA Editorial. 2008. ISBN(13): 978-84-7897-887-8.

[3] SecureW2: http://www.securew2.com

[4] Open1x: http://open1x.sourceforge.net

[5] Network Manager: http://projects.gnome.org/NetworkManager/?x=22

[6] Wpa_supplicant: http://hostap.epitest.fi/wpa_supplicant

[7] Catalyst 3550 Multilayer Switch Software Configuration Guide: http://www.cisco.com/en/US/docs/switches/lan/catalyst3550/software/release/12.1_8_ea1/configuration/guide/SwCfg.pdf

[8] MAC Authentication Bypass: http://www.cisco.com/univercd/cc/td/doc/solution/macauthb.pdf

[9] Hassell, Jonathan. *RADIUS: Securing Public Access to Private Resources*. O'Reilly. 2002. ISBN(10): 0-596-00322-6

[10] *RFC 2989:* Criteria for Evaluating AAA Protocols for Network Access: http://tools.ietf.org/html/rfc2989

[11] RFC 2865: *Remote Authentication Dial In User Service (RADIUS):* http://www.ietf.org/rfc/rfc2825.txt

[12] RFC 2866: *RADIUS Accounting: http://www.ietf.org/rfc/rfc2866.txt*

[13] *TACACS+ and RADIUS Comparison:* http://www.cisco.com/application/pdf/paws/13838/10.pdf

[14] *The TACACS+ Protocol*: http://tools.ietf.org/html/draft-grant-tacacs-02

[15] RFC 3588: *Diameter Base Protocol: http://www.ietf.org/rfc/rfc3588.txt*

[16] freeRADIUS: http://www.freeradius.org

[17] Cisco ACS:

http://www.cisco.com/en/US/prod/collateral/vpndevc/ps5712/ps2086/product_bulletin_c25-453313.pdf

[18] Microsoft IAS: http://technet.microsoft.com/en-us/library/cc958027.aspx

[19] Juniper SBR: http://www.juniper.net/us/en/local/pdf/datasheets/1000268-en.pdf

[20] GNU RADIUS: http://www.gnu.org/s/radius

[21] OpenRADIUS: http://evbergen.home.xs4all.nl/openradius

[22] BSD RADIUS: http://www.bsdradius.org

[23] TekRADIUS: http://www.tekradius.com

[24] OpenDiameter: http://diameter.sourceforge.net

[25] freeDiameter: http://www.freediameter.net/trac

[26] Eduroam: http://www.eduroam.org

[27] *Sistema para el control de acceso a red basado en servicios* - Universidad del País Vasco / Euskal Herriko Unibersitatea (UPV/EHU): http://www.rediris.es/jt/jt2009/ponencias/VirtII-3.ppt

[28] *Proyecto de Integración de la Tarjeta Inteligente en UPNs* - Universitat de les Illes Balears (UIB): http://www.rediris.es/jt/jt2003/archivo-jt/SALAB/06112003/sesionVIII/MBordoy-JT03.ppt

[29] *Autenticación centralizada mediante CAS y federación de servicios* - Barcelona Supercomputing Center (BSC): http://www.rediris.es/jt/jt2010/ponencias/jt2010-jt-serv_feder_2-2.pdf

[30] Deploying RADIUS: http://deployingradius.com

Annex I: freeRADIUS

Aquest annex és un petit manual sobre freeRADIUS [16] [30], que pot ser útil per qualsevol persona que vulgui iniciar-se en aquest servidor AAA. Aquesta guia està dividida en tres parts:

Introducció Es fa una breu introducció a freeRADIUS i es mostren les seves principals característiques.

Fitxers de freeRADIUS S'explica quina funció té cadascun dels principals fitxers de freeRADIUS i la forma en que es configuren.

Unlang Es fa una breu introducció sobre aquest llenguatge de programació propi de freeRADIUS, útil per configurar els fitxers.

Introducció a freeRADIUS

FreeRADIUS és un dels servidors RADIUS més modular i amb més funcionalitats d'avui en dia (incloent servidors gratuïts i de pagament). Va ser creat l'any 1999 per un grup de desenvolupadors amb més d'una dècada d'experiència en implementació i desenvolupament de software RADIUS, en enginyeria del software i en la gestió de paquets Unix. El producte és el resultat de la unió entre diferents experts desenvolupadors de RADIUS, incloent alguns desenvolupadors del sistema operatiu Debian GNU/Linux. Des de la versió 2, freeRADIUS té llicència GNU GPL.

Les principals característiques de freeRADIUS són:

- Funcionament en servidors Unix (Linux i Solaris) i en arquitectures x86 i x64.
- Suport a una gran varietat de bases de dades: MySQL, MS SQL, Oracle i Postgres.
- Suport per als principals serveis de directori: LDAP i Active Directory.
- Suport per a varis mètodes d'autenticació: PAP, CHAP, MSCHAPv1, MSCHAPv2, SIP, EAP-MD5, EAP-SIM, EAP-TLS, EAP-TTLS, EAP-PEAP i LEAP.
- Suport per a tots els atributs de RADIUS definits en els seus RFCs [11] [12].
- Suport per a atributs RADIUS propietaris (VSA) de BinTec, Foundry, Cisco, Juniper, Lucent/Ascend, HP Procurve, Microsoft, USR/3Com i Acc/Newbridge entre d'altres.
- Assignació de IPs basat en IP pools.
- Funcionament com a servidor RADIUS o com a proxy.

- Ús de la característica *fall through*, que permet que les configuracions específiques d'usuari siguin ignorades en benefici de les configuracions per defecte.
- Limitació del nombre de connexions simultànies.
- Agrupació d'usuaris en *huntgroups* (segons el client o el port físic del client als quals estan connectats).
- Ús de *hints*, que permeten escollir les polítiques d'autorització que se li ha d'aplicar a l'usuari en funció de la sintaxi del seu nom d'usuari.
- Suport per a VMPS.
- Llenguatge *unlang* per a la personalització de totes les fases AAA.
- Suport per DHCP.

El fitxers de freeRADIUS

En aquest apartat es descriuen les funcionalitats dels principals fitxers de freeRADIUS. Tots els fitxers de configuració es troben en el directori $FREERADIUS_DIR*/etc/raddb/* i els seus subdirectoris.

El fitxer radiusd.conf

Radiusd.conf és el fitxer principal de configuració de freeRADIUS. En ell es defineixen les principals directius per fer funcionar el servidor, com per exemple el usuari sota el qual correrà el servidor, el port per on ha d'escoltar les peticions, etc.

La gran part de la resta de fitxers de configuració són cridats per aquest fitxer utilitzant directives *include*.

Els servidors virtuals

Inicialment freeRADIUS permetia una política de configuració global per a totes les peticions rebudes.

Des de la versió 2, es permet la creació de servidors virtuals. Un servidor virtual funciona com un servidor RADIUS convencional. Per tant, mitjançant servidors virtuals, freeRADIUS pot executar varis servidors RADIUS al mateix temps.

En els servidors virtuals es configuren les regles d'autenticació, autorització i *accounting*. Tenir diferents servidors virtuals permet definir varies configuracions d'autenticació, autorització i *accounting* en funció, per exemple, de l'adreça IP del client.

Cada servidor virtual té el seu propi fitxer de configuració que es troba en el directori *sites-available*. En aquest directori es troben els fitxers de configuració tant dels servidors virtuals actius com dels inactius. Els servidors virtuals actius tenen un enllaç simbòlic del directori *sites-available* cap al directori *sites-enabled*. Per tant, si es vol habilitar un servidor virtual caldrà crear un enllaç simbòlic des del directori *sites-enabled* al directori *sites-available*. Per altra banda, si és vol inhabilitar un servidor virtual, només cal eliminar l'enllaç simbòlic del directori *sites-enabled*.

A continuació es mostra un exemple molt senzill de servidor virtual on s'inclouen les configuracions d'autenticació, autorització i *accounting*:

```
authorize {
        preprocess
        ldap
        files
}
authenticate {
        Auth-Type LDAP {
                ldap
        }
}
accounting {
        sql
}
```

Com s'observa, aquest servidor està dividit en tres parts: *authorize*, *authenticate* i *accounting*.

Els mòduls de cada secció s'executen per ordre. En la secció *authorize*, el servidor freeRADIUS processarà els mòduls *preprocess* (busca entrades de l'usuari en els fitxers *hints* i *huntgroups*), *ldap* (busca entrades de l'usuari en l'LDAP) i files (busca entrades de l'usuari en el fitxer users).

En la secció *authenticate*, s'està indicant que s'utilitzi l'LDAP per autenticar els usuaris en el sistema. El paràmetre *Auth-Type LDAP* s'assigna automàticament si el servidor ha trobat una entrada de l'usuari en l'LDAP en l'execució del mòdul *ldap* de la secció *authorize*. Per tant, si en el mòdul *ldap* de la secció *authorize* no s'ha trobat l'usuari, el mòdul *ldap* de la secció *authenticate* no s'executarà.

En la secció *accounting* s'ha afegit el mòdul *sql*, el qual s'encarrega d'emmagatzemar la informació en una base de dades.

El fitxer *users*

En el fitxer *users* es poden crear els usuaris del sistema i permet realitzar les fases d'autenticació i autorització.

Les entrades d'usuaris són processades en l'ordre en el que apareixen en el fitxer. Un cop l'entrada coincideix amb l'usuari que vol obtenir accés a la xarxa, el processament del fitxer s'atura. Si es desitja que no s'aturi al trobar una coincidència es pot definir el paràmetre *Fall-Through = Yes*.

Existeix un usuari especial anomenat DEFAULT. Sempre que el servidor freeRADIUS trobi una entrada per aquest usuari la processarà. Per tant, DEFAULT identifica tots els usuaris del sistema. En el fitxer *users* es poden definir varies entrades per usuari, incloent el DEFAULT. Si volem que freeRADIUS tracti totes les entrades per l'usuari DEFAULT, caldrà definir el paràmetre *Fall-Through = Yes* en totes les entrades DEFAULT menys l'última.

Tot i que en les primeres versions de freeRADIUS tot usuari del sistema havia d'estar definit en aquest fitxer, actualment no es recomana el seu ús per a realitzar la fase d'autenticació. El principal motiu per evitar autenticar mitjançant el fitxer *users* és que les contrasenyes s'emmagatzemen en text pla. En el seu lloc, es recomana utilitzar una base de dades, un servei de directori o l'autenticació local de Unix per realitzar la fase d'autenticació.

En canvi, el fitxer *users* és un bon lloc per configurar les polítiques d'autorització. En ell podem definir quins permisos de xarxa volem donar a cada usuari (o grups d'usuaris) d'una forma còmoda.

A continuació es mostra un extracte qualsevol del fitxer de *users*:

```
steve           Cleartext-Password := "testing"
                Service-Type = Framed-User,
                Framed-Protocol = PPP,
                Framed-IP-Address = 172.16.3.33,
                Framed-IP-Netmask = 255.255.255.0,
                Framed-Routing = Broadcast-Listen,
                Framed-Filter-Id = "std.ppp",
                Framed-MTU = 1500,
                Framed-Compression = Van-Jacobsen-TCP-IP
```

```
DEFAULT         Service-Type == Framed-User
                Framed-IP-Address = 255.255.255.254,
                Framed-MTU = 576,
                Service-Type = Framed-User,
                Fall-Through = Yes

DEFAULT         Framed-Protocol == PPP
                Framed-Protocol = PPP,
                Framed-Compression = Van-Jacobson-TCP-IP
```

En l'exemple anterior es mostren tres entrades, una per l'usuari *steve* i dos per l'usuari DEFAULT. La primera línia de cada entrada indica les condicions que ha de complir la connexió de l'usuari per a que freeRADIUS tracti l'entrada.

La resta de línies de cada entrada fan referencia als *reply-items*. Els *reply-items* són els atributs RADIUS o VSA que el servidor freeRADIUS enviarà al client RADIUS i que aquest aplicarà sobre la connexió de l'usuari. Dit d'una altra forma, el *reply-items* indiquen les polítiques d'autorització que s'han d'aplicar a l'usuari.

Per tant, en aquest exemple de fitxer *users* s'ha definit l'usuari *steve* i les seves polítiques d'autorització. Per a que es puguin aplicar aquestes polítiques caldrà que la contrasenya introduïda per l'*steve* sigui *testing*. Notar que això és la fase d'autenticació.

Les dos entrades DEFAULT comproven el tipus de servei que sol·licita l'usuari (atribut *Service-Type*) i el tipus de trama de protocol que s'utilitzarà en l'accés (atribut *Framed-Protocol*) respectivament.

Es pot observar que en la última línia de l'entrada del primer usuari DEFAULT es troba l'atribut *Fall-Through = Yes*. Això vol dir que encara que aquesta entrada hagi estat tractada pel servidor freeRADIUS, el processament del fitxer continuarà i l'entrada del segon usuari DEFAULT podrà ser també tractada (sempre que es compleixin les condicions).

En canvi, l'usuari *steve* no té habilitat aquest paràmetre. Quan el servidor acabi de tractar l'entrada de l'usuari *steve* (si es compleixen les condicions) el processament del fitxer *users* acabarà, i les entrades DEFAULT no seran tractades.

El fitxer *clients.conf*

En aquest fitxer es defineixen els clients RADIUS que es vol que es comuniquin amb el servidor. Cada client RADIUS tindrà una entrada dedicada en aquest fitxer.

Un exemple d'entrada per al fitxer clients seria:

```
client 192.168.100.100 {
        secret = testing123
        shortname = cisco1
        nastype = cisco
}
```

En aquest exemple s'ha creat una entrada pel client RADIUS amb adreça IP 192.168.100.100. S'ha indicat el *shared secret* que comparteix amb el servidor freeRADIUS, un nom amistós (cisco1) per poder-lo diferenciar dels altres clients i el nom del fabricat (*nastype*), que ajuda al servidor a saber amb quin tipus de client s'està comunicant, i per tant, quins VSA es poden utilitzar.

El fitxer *proxy*

Aquest fitxer és utilitzat per configurar un servidor freeRADIUS com a *proxy* d'altres servidors freeRADIUS.

El fitxer *eap.conf*

Aquest fitxer es utilitzat per configurar els mètodes EAP que volem que suporti el servidor freeRADIUS. En ell es configura l'EAP-Method per defecte que es vol que utilitzi el servidor freeRADIUS i els EAP-Methods alternatius que el servidor també acceptarà.

Això vol dir que el servidor intentarà arribar a un acord amb el *Supplicant* per utilitzar l'EAP-Method que té configurat defecte. Si el *Supplicant* rebutja l'oferiment, farà la seva proposta de EAP-Method. El servidor freeRADIUS acceptarà sempre i quan l'EAP-Method proposat pel *Supplicant* sigui un dels configurats com a alternatiu en el servidor.

El fitxer *hints*

El fitxer *hints* es utilitzat per donar pistes al servidor RADIUS sobre quins serveis ha d'oferir a un *Supplicant* en funció del seu nom d'usuari.

Suposem que s'ha configurat un usuari per a que per defecte sigui col·locat en l'VLAN 10. Però en determinades situacions, es vol que aquest usuari es col·loqui en l'VLAN 20. Per a indicar-li al servidor que vol ser col·locar a l'VLAN 20, l'usuari podria utilitzar el sufix ".20" en el seu nom d'usuari.

A continuació es mostra com seria l'entrada del fitxer *hints*:

```
DEFAULT        Suffix == ".20", Strip-User-Name = Yes
               Hint = "VLAN.20",
```

Utilitzant aquesta entrada s'han associat tots els noms d'usuari que acaben amb la cadena ".20" al *hint* VLAN.20.

L'atribut freeRADIUS *Strip-User-Name = Yes* serveix per eliminar el sufix ".20" del nom d'usuari a l'hora d'autenticar (en la BBDD o en el servei de directori el nom d'usuari estarà emmagatzemat sense el sufix). El següent pas a realitzar seria afegir la següent entrada al fitxer *users*:

```
DEFAULT        Hint == "VLAN.20"
               Tunnel-Type = "VLAN",
               Tunnel-Medium-Type = "IEEE-802",
               Tunnel-Private-Group-Id = "20"
```

Amb aquesta entrada en el fitxer *users* s'estan associant tots els usuaris etiquetats amb el hint VLAN.20 a l'VLAN 20. Els atributs RADIUS *Tunnel-Type, Tunnel-Medium-Type* i *Tunnel-Private-Group-Id* serveixen per indicar al client RADIUS l'VLAN on col·locar l'usuari.

El fitxer *huntgroups*

Un *huntgroup* es un conjunt de clients RADIUS o ports de clients RADIUS. Mitjançant aquest fitxer, freeRADIUS permet agrupar els usuaris segons el *huntgroup* al qual estan connectats.

Suposem que es desitja que tots els usuaris connectats a través del client amb adreça IP 192.168.100.10 pertanyin a l'VLAN 30. Es podria afegir la següent entrada al fitxer *huntgroups*:

```
Huntgroup1      NAS-IP-Address == 192.168.100.10
                Group = client1
```

Ara, tots els usuaris que es connectin des del client 192.168.100.10 formaran part del grup *Hountgroup1*. Per indicar les polítiques d'autorització per als membres del grup es pot afegir una entrada com la següent al fitxer *users*:

```
DEFAULT         Huntgroup-Name == "client1"
                Tunnel-Type = VLAN,
                Tunnel-Medium-Type = IEEE-802,
                Tunnel-Private-Group-Id = 30
```

Amb aquesta configuració, tots els usuaris que es connectin a través del client amb IP 192.168.100.10 seran col·locats a l'VLAN 30.

El fitxer *policy.conf*

En el fitxer *policy.conf* es defineixen subrutines que poden ser cridades des de qualsevol fitxer de configuració. La programació d'aquestes subrutines es realitza mitjançant *unlang*, el llenguatge de programació de freeRADIUS.

El fet de poder crear subrutines pròpies permet una gran personalització del servidor.

Els fitxers *dictionary*

En els fitxer de diccionari es troben tots els atributs RADIUS i VSA suportats per freeRADIUS.

Cada fitxer fa referència a un grup d'atributs. Per exemple, en el fitxer dictionary.rfc2865 es defineixen els atributs RADIUS descrits en l'RFC 2865 [11] i en el fitxer *dictionary.cisco* es defineixen els atributs VSA de Cisco.

El fitxer *sql.conf*

En el fitxer *sql.conf* es configuren tots els paràmetres relacionats amb la base de dades (quin gestor de base de dades s'utilitza, quins són els paràmetres de connexió, etc.).

Les plantilles de les bases de dades

FreeRADIUS facilita una plantilla (*schema*) per cada gestor de base de dades suportat. Aquestes plantilles són les encarregades de crear la bases de dades freeRADIUS i les seves taules. Les plantilles poden ser modificades per personalitzar el sistema.

El fitxer *dialup.conf*

Aquest fitxer conté les consultes SQL que es realitzaran sobre les bases de dades. El fitxer *dialup.conf* pot ser modificat per personalitzar les consultes.

El fitxer *ldap*

En aquest fitxer es realitza la configuració del mòdul *ldap*.

El fitxer *mschap*

Aquest fitxer permet configurar l'autenticació d'usuaris en Active Directory utilitzant el mètode d'autenticació MSCHAP. En el fitxer s'indica la localització del binari *ntlm_auth*, que serà l'encarregat d'enviar les peticions d'autenticació a *l'Active Directory*.

El directori de certificats

freeRADIUS proporciona la funcionalitat creació de certificats de servidor, de client i d'Autoritat Certificadora. Aquestes eines es troben en el directori de freeRADIUS *etc/raddb/certs*.

freeRADIUS facilita una plantilla per a la creació d'aquests certificats. Quan el *daemon* de freeRADIUS s'executa per primer cop, executa l'script *bootstrap* que crea els certificats a partir de les plantilles. Els certificats de servidor i client són signats per l'Autoritat Certificadora creada.

Es poden modificar els certificats editant les plantilles. Dins del mateix directori existeix un *make* que permet crear el certificats desitjat amb el format que es vulgui. Els formats de certificats disponibles són *pem* i *der*.

Unlang: el llenguatge de freeRADIUS

Com a part de la política de freeRADIUS de ser un servidor potent i flexible es va crear un llenguatge de programació propi aplicable als seus fitxers de configuració.

Unlang és un llenguatge simple de processament. El seu objectiu és facilitar la feina a l'hora d'implementar polítiques d'autenticació i autorització.

Unlang permet referenciar variables, però no permet crear-ne de noves. Les variables referenciades podran ser qualsevol atribut dels diccionaris (atributs RADIUS o VSA) o variables pròpies de freeRADIUS.

Per referenciar una variable s'utilitzarà la següent sintaxi *%{nom-variable}*, per exemple: *%{User-Name}*.

Aquest llenguatge també permet l'ús de l'operador condicional *if*. Aquest operador té el mateix funcionament que en altres llenguatges de programació: si és compleix la condició que el precedeix executa el codi que es troba dins del bloc. *Unlang* també permet l'ús dels operadors *else* i *elsif*:

```
If (condició) {
        instruccions
```

```
}
elsif (condició) {
        instruccions
}
else {
        instruccions
}
```

Normalment, *unlang* s'utilitza per fer comprovacions sobre les trames rebudes o per fer modificacions sobre els atributs a enviar en les trames de resposta. Un exemple de codi en *unlang* per modificar un atribut seria el següent:

```
if ("%User-Name" == 'foo') {
        update reply {
                Reply-Message = "Hello foo"
        }
}
```

En aquest exemple s'està modificant el valor de l'atribut RADIUS *Reply-Message*, que permet mostrar un missatge personalitzat a l'usuari. Per exemple, Si l'usuari s'anomena *foo,* rebrà el missatge *Hello foo*. L'assignació d'atributs es realitza sempre dins d'un bloc Update.

Unlang també és útil per diferenciar les peticions rebudes mitjançant 802.1x i MAB. Les trames 802.1x van encapsulades en EAP mentre que les trames MAB no. Per tant, freeRADIUS podrà saber si ha rebut una trama 802.1x o MAB comprovant si la trama rebuda és EAP:

```
if (EAP-Message) {
        #Trama 802.1x
}
else {
        #Trama MAB
}
```

Annex II: Glossari

AAA - Authentication, **A**uthorization and **A**ccounting.

ACS - **A**ccess **C**ontrol **S**erver.

ASCII - **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange.

ARAP - **A**pple **R**emote **A**ccess **P**rotocol.

ASF - **A**lert **S**tandard **F**ormat.

AVP - **A**ttribute **V**alue **P**air.

BIOS - **B**asic **I**nput/**O**utput **S**ystem.

BSC - **B**arcelona **S**upercomputing **C**enter.

CESCA - **Ce**ntre de **S**upercomputació de **Ca**talunya.

CHAP - **C**hallenge **H**andshake **A**uthentication **P**rotocol.

CPD - **C**entre de **P**rocessament de **D**ades.

DHCP - **D**ynamic **H**ost **C**onfiguration **P**rotocol.

DNS - **D**omain **N**ame **S**ystem.

EAP - **E**xtensible **A**uthentication **P**rotocol.

EAPOL - **E**xtensible **A**uthentication **P**rotocol **O**ver **L**AN.

Eduroam - **Edu**cation **Roam**ing.

FAST - **F**lexible **A**uthentication via **S**ecure **T**unneling.

FIB - **F**acultat **I**nformàtica de **B**arcelona.

FLR - **F**ederation-**L**evel **R**ADIUS Servers.

GINA - **G**raphical **I**dentification and **A**uthentication.

GSM - **G**lobal **S**ystem for **M**obile Communications.

GTC - **G**eneric **T**oken **C**ard.

IANA - **I**nternet **A**ssigned **N**umbers **A**uthority.

IAS - **I**nternet **A**ccess **S**erver.

IdP - **Id**entity **P**roviders.

IEEE - **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers.

IETF - **I**nternet **E**ngineering **T**ask **F**orce.

IP - **I**nternet **P**rotocol.

LAN - **L**ocal **A**rea **N**etwork.

LCFIB - **L**aboratori de **C**àlcul de la **F**acultat **I**nformàtica de **B**arcelona.

LDAP - **L**ightweight **D**irectory **A**ccess **P**rotocol.

LEAP - **L**ightweight **E**xtensible **A**uthentication **P**rotocol.

MAB - **M**AC **A**uthentication **Bypass**.

MAC - **M**edia **A**ccess **C**ontrol.

MD5 - **M**essage **D**igest **5**.

MITM - **M**an **I**n **T**he **M**iddle.

MSCHAP - **M**icro**S**oft **C**hallenge **H**andshake **A**uthentication **P**rotocol.

MSDNAA - **M**icro**s**oft **D**eveloper **N**etwork **A**cademic **A**lliance.

NAK - **N**egative-**A**c**k**nowledge Character.

NAS - **N**etwork **A**ccess **S**erver.

NASI - **N**etWare **A**synchronous **S**ervice **I**nterface.

NT/LM – **NT L**AN **M**anager.

OSI - **O**pen **S**ystems **I**nterconnection.

OTP - **O**ne **T**ime **P**assword.

PAC - **P**rotected **A**ccess **C**redentials.

PAM - **P**luggable **A**uthentication **M**odules.

PAP - **P**assword **A**uthentication **P**rotocol.

PEAP - **P**rotected **E**xtensible **A**uthentication **P**rotocol.

PPP - **P**oint-to-**P**oint **P**rotocol.

QoS - **Q**uality **O**f **S**ervice.

RADIUS - **R**emote **A**uthentication **D**ial **I**n **U**ser **S**ervice.

RFC - **R**equest **F**or **C**omments.

SAI - **S**istema d'**A**limentació **I**ninterrompuda.

SBR - **S**teel-**B**elted **R**ADIUS.

SCTP - **S**tream **C**ontrol **T**ransmission **P**rotocol.

SHA - **S**ecure **H**ash **A**lgorithm.

SIM - **S**ubscriber **I**dentity **M**odule.

SNMP - **S**imple **N**etwork **M**anagement **P**rotocol.

SP - **S**ervice **P**roviders.

TACACS(+) - **T**erminal **A**ccess **C**ontroller **A**ccess-**C**ontrol **S**ystem.

TCP - **T**ransmission **C**ontrol **P**rotocol.

TLR - **C**onfederation Top-**L**evel **R**ADIUS Servers.

TLS - **T**ransport **L**ayer **S**ecurity.

TTLS - Tunneled **T**ransport **L**ayer **S**ecurity.

UIB - **U**niversitat de les **I**lles **B**alears.

UDP - **U**ser **D**atagram **P**rotocol.

UMTS - **U**niversal **M**obile **T**elecommunications **S**ystem.

UPN - **U**ser **P**ersonalized **N**etwork.

UPV / EHU - **U**niversidad del **P**aís **V**asco / **E**uskal **H**erria **U**nibersitatea.

VLAN - **V**irtual **L**ocal **A**rea **N**etwork.

VMPS - **V**LAN **M**anagement **P**olicy **S**erver.

VQP - **V**LAN **Q**uery **P**rotocol.

VSA - **V**endor **S**pecific **A**ttribute.

XTACACS - E**x**tended **T**erminal **A**ccess **C**ontroller **A**ccess-**C**ontrol **S**ystem.